A Lightweight Solution for Remote Debugging of Microcontroller Projects

Jannie Li

Abstract

The remote debugging of breadboards with microcontrollers and circuits, especially in teaching, is very difficult with currently available tools, which makes educating students new to using this hardware difficult and reliant on in-person office hours. This can be a limited service, so students may not be available at the time instructors are or vice versa. Furthermore, as the current coronavirus pandemic shows, options are necessary for educating new hardware users on how to debug their circuits without the instructor and student both being in the same room. Thus, this project aims to provide a lightweight solution to helping students debug circuits remotely. The solution is heavily inspired by Heimdall, a research project that was underway but required a lightweight alternative during the current crisis. We demonstrate how a lightweight solution using existing screen sharing software, a student's phone or camera, and a 3D printed stand can enable instructors to help students debug their circuits remotely.

Introduction

The importance of learning how to use electronics, circuits, and embedded systems remotely has only been increasing. More people have begun learning about microcontroller projects as a hobby, through online education like MOOCs, and during the current COVID-19 pandemic, through virtual classrooms. This has highlighted a necessity for remote debugging that is accessible for students new to the technology.

Without any additional resources, students often attempt to debug their hardware through remote help, with either an instructor or a friend. This often means either sending pictures of their circuit setup or through video sharing platforms. In trying out these solutions, we discovered several major problems from both the students' and instructors' ends.

An instructor will often attempt to perform various steps to check the student's project before being able to offer solutions. Often in order, these will be:

1. **Making sure wires are inserted into the breadboard all the way.** Issues with this include being able to properly see the breadboard and confirm that the student has done this.
2. **Making sure wires are in the correct place.** Issues with this include difficulty getting a clear view of the student's breadboard, due to either faulty connections or camera quality. To get a truly clear enough image to perhaps understand the orientation of sensors or to read microcontroller text, a student must hold the camera device extremely still. New

students will also have difficulty following instructor directions on what an instructor needs to see and will need to provide multiple angles of high quality images to help an instructor properly assess the breadboard. Instructors will have difficulty “pointing” and a student trying to point to or show a certain part of their circuit will often interrupt the instructor’s view.

3. **Making sure components are getting correct power.** This would be done with a multimeter, which students may not possess or know how to properly use. Again, difficulties with teaching students how to use the multimeter will compromise instructor views of wiring and the student will struggle with juggling showing the instructor their circuit and multimeter while also trying to properly use the multimeter.
4. **Actively testing the student’s system.** Instructors may want to actively test a circuit to gain a better understanding of the wiring. This requires a student to juggle the camera view, which is already difficult to achieve, with responding to instructor requests that can be very confusing. Instructors will have difficulty knowing how to communicate changes and testing protocol. This leads to further visual/technical confusion and can take a very long time, which frustrates both student and instructor.
5. **Checking that a student’s written code is correct.** This can be done through a student’s screen sharing their code, but will also often require views of the circuit to test and adjust behavior. Juggling this is, again, a complicated and frustrating process.

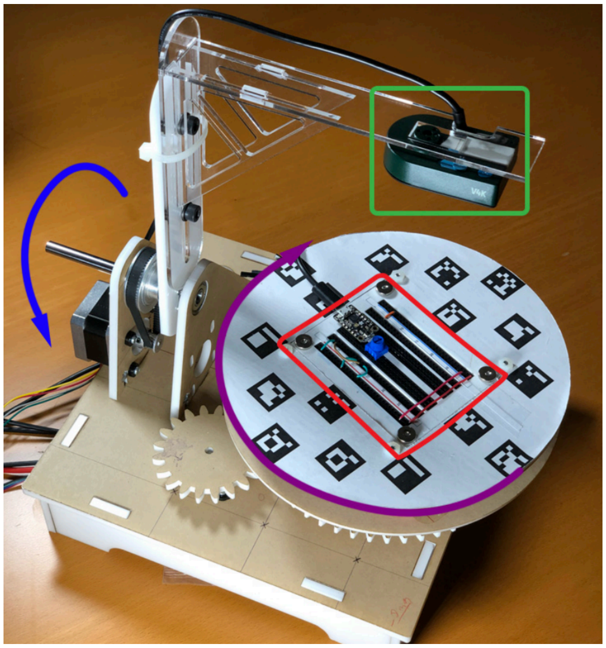
This puts a lot of pressure on both student and instructor patience and also removes the familiarity of human behavioral cues that can aid both to understand how the other is assessing and understanding a given problem. The difficulty of each step of the process can also lead instructors to try similar things multiple times, again making the entire process hard to manage and taking up time that could be used to help other students.

To address these problems, we created a workflow integrating existing tools that students are already using to take online classes as well as lightweight additions that could be added to their hardware toolkits that will provide a much better option for remote debugging.

Related Work

The increase in remote learning has prompted a variety of products, ranging from augmented breadboards that provide visualizations of voltages and flow to full simulations students can experiment with on their computer screens. However, it is still essential that at some point students get access to the actual physical systems and embodied experiences that they would use to build more complex projects and move on to more complex hardware. Being able to build with regular breadboards, wires, and microcontrollers also puts students in conversation with the variety of online forums, articles, and other resources that hobbyists and professionals frequently use. This leaves us with the same problem as before: how can a student physically present with their project get debugging help remotely?

The research project Heimdall offers a solution to remote debugging where an instructor is not present. Heimdall proposes a system that enables a robotic gantry to take pictures of a student breadboard at multiple angles. Computer vision software then compiles these 100+ images into a 3D view of the breadboard. An instructor connected to the server on which this is uploaded can then view the breadboard almost as if they were personally in the room—from any angle they wish with a clear image. Furthermore, the instructor is able to perform virtual voltage experiments, as with a multimeter and simulate modifications by isolating portions of the breadboard to inject a different signal. Thus, the user interface presented to the instructor is far more comprehensive and easy to use than the shaky phone/camera a student would normally resort to. Because the system saves this information, it can also be seen over time, giving both student and instructor time to properly assess changes and present their replies, removing the frustrating back and forth process necessary over video sharing services.



A photo of Heimdall

The Heimdall system was created to be available in a university lab after hours, enabling students to get help even when instructors aren't present. However, due to the COVID-19 pandemic, as well as the proliferation of courses that are exclusively virtual and online, this system is limited by the sharing of space and machinery.

This project presents a lightweight alternative that thus builds on Heimdall's idea, but enabling it to be used by students who are unable to access a university lab, and can get similar benefits from their homes.

Design Rationale

Students taking an online hardware class could take advantage of existing tools and methods that include:

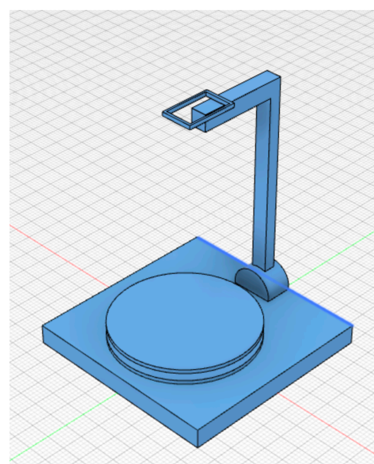
- A hardware box shipped to their homes
- An existing familiarity with video and screen sharing devices such as Zoom, which they will already be using to attend class
- Offering remote help at different time zones since TAs may find those hours more convenient
- Recordings of lectures or complete breadboards that can be made available by an instructor

Our proposed solution was designed using a combination of these existing tools because we wanted to make the addition of a debugging system as simple as possible, adding the least amount of additional hardware and software to the online learning system, which is already often difficult to navigate and complex. Thus, we focused primarily on creating a workflow using existing tools and added new tools only when strictly necessary.

This led us to the following workflow:

1. Student joins instructor office hours and shares a live camera feed of their breadboard using a mechanical gantry stand
2. Instructor helps student debug, using live debugging methods, including having the student change the camera angle for a better view and having the student make and test modifications
3. Instructor is able to demonstrate and further explain concepts using Zoom annotation features
4. Student and instructor are able to debug the system
5. Student can also refer back to a pre-recorded explanation from the professor to confirm understanding of key concepts

Implementation



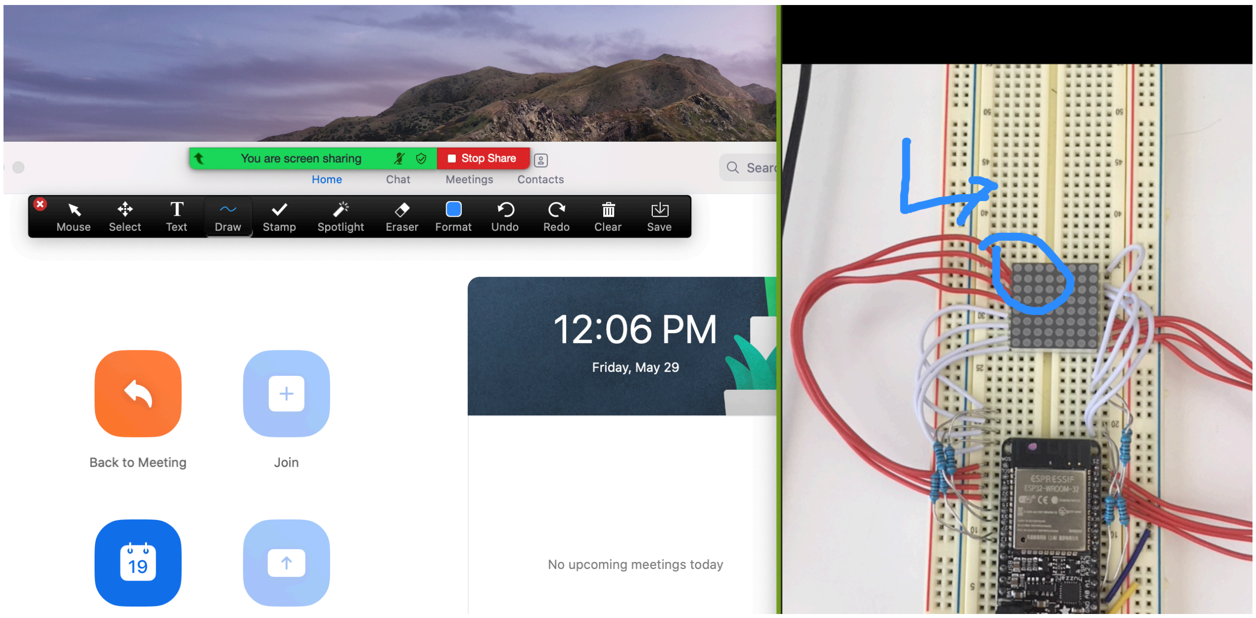
Visual Inspection

A mechanical gantry device will be delivered to each student in their hardware box. It will have two rotational axes, similar to the Heimdall gantry. The surface where students can place their breadboard will be able to rotate circularly. The stand holding the phone will be able to lower the camera to the level as needed. A phone clip at the top will secure the student's phone so the phone's camera can be used.

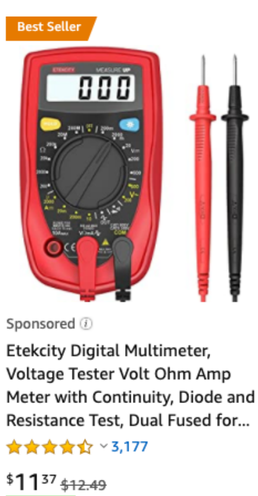
Zoom Interface

Zoom enables users to write directly on shared screens using the annotation feature. Unfortunately, this does not include mobile shared screens. However, both Android and Apple mobile OS systems enable users to share their screens on their laptops by manually plugging in their machines or by using apps. A good option for iPhone users is plugging their phone into their computers and using QuickTime to use their camera feed. An option for Android users includes using the app DroidCam. There are many other options that students may prefer.

Once they share their camera feed with their computer screens, they can use Zoom's screen share feature to share the view of their breadboard with the instructor. This then enables instructors to draw, highlight and otherwise mark up a view of the breadboard, helping students understand where to focus their efforts. Students will also be able to share their IDE screens for easy help with code. Remote control features in Zoom enable instructors to access student code directly as well.



Furthermore, the constant sharing of screens also enables the student to point to confirm and allows the instructor and student to continue seeing each other to understand where the other person is at in understanding the given issue. Often, remote debugging help, such as Heimdall's interface and forums are not time synchronized, which means that students and instructors lose valuable cues such as head nods, eye contact, and more. By utilizing these capabilities in face-sharing and screen-sharing applications, users can gain a more human element to their work.

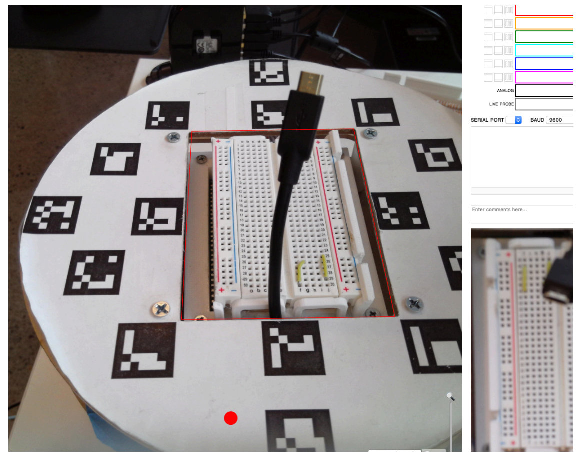


<u>Measurement</u>

A simple multimeter can be as cheap as about $12, which can be a great option for shipping to students. This will enable to not miss out on learning how a multimeter works and also get on-hand TA help with navigating its use with the stand and Zoom interface. This also provides measurement data as needed during debugging sessions and can be shown on camera as well.

Remote Access to a Working Demo

Since many instructors will be recording their lectures for the benefit of students in various time zones, students may be able to access a recording of a working demo and explanations while working on projects. Furthermore, an instructor could post a 3D interactive recording using Heimdall on a shared server that students can access after hours. This could help students explore a live demo on their own as well, testing various voltages and viewing the breadboard from various angles.



Camera Alternatives

Some students may not have phones to use or struggle with using their phone cameras. In this case, especially for more experienced students, an educational alternative is possible. For example, since students will already be familiar with the ESP32, they can use a camera that can be connected with the Arduino directly, sharing its feed with their computers. One cheap example would be the M5Stack ESP32 camera module, available for under $10. Drawbacks to this alternative include a lower frame rate, but the stationary nature of the breadboard can make this a viable alternative.

User Experience

The following is an example workflow to help clarify exactly how the system would be used in context.

Nick needs help with wiring his potentiometer. He's not sure why it's not sending signals to his microcontroller and thus, not showing up in the Serial Monitor of his Arduino IDE. He finds out through the class lecture and class website that there will be office hours on Monday afternoon with TA Amy. On Monday afternoon, he joins her Zoom call. He has never used the debugging system included in his hardware kit yet, but Amy shows him what it looks like by showing him her own and explains that the stand is for his phone and the multimeter measures how much voltage is going through his wires. He is able to easily set up the system by using a screen mirroring system he was advised to download at the beginning of the semester and placing his phone on the stand, clipping it into place. He puts his breadboard on the rotating bed of the stand and sees it on his phone's mirrored screen on his laptop. He knows how to use Zoom's screen share feature and is able to share that screen with Amy. Amy asks him to rotate and move his camera to a spot that helps her best understand the problem. This takes a few minutes for both of them to get oriented, but after about five minutes, Amy understands the problem— Nick has confused his ground and power lines and reversed them.

Amy tells Nick to show her the part of his breadboard where he connects the potentiometer with ground and power sources. He is now familiar enough with the mechanical arm and rotating base to easily do this. Amy begins annotating his screen, showing him which of the three wire possibilities should be connected to ground and power and what he actually did. Nick nods to indicate he understands and Amy can see that in his camera view. She then watches as he fixes the wiring and reruns his Serial Monitor. He shares his Serial Monitor screen with her to demonstrate it working.

Later that night, Nick is working on adding another sensor to his project, and goes on the class website to view a pre-recorded interactive 3D view of an example project. His instructor had prepared this demo to help students with understanding that day's lecture since the students could not go to the front of the room to check the actual physical boards. Nick finds it easy to navigate through the Heimdall interface and gains a better understanding, which he uses to successfully complete his project.

Limitations and Future Work

While this proposed system clearly improves upon the shaky footage and unstandardized office hours procedures, there is still much more work to do to further improve it.

Improving Pointing

Using Zoom's annotation features is helpful when the problem is somewhat simple, but if a student adjusts the camera view, the annotations do not move accordingly and begin to block newer views. The student is able to physically point to parts of the board on camera since they are using a live view, but the instructor is unable to do the same. Thus, a pointing feature would be very helpful when using live feeds. This could be done in the future using a simple laser pointer that could be attached to the gantry structure. However, that would require the entire system to be connected to a software interface and a server. Another possibility, also requiring a software interface, would be an annotation software that tracks the angle and distance, keeping the annotation where it was originally written when the user moves the camera. For an immediate lightweight solution, we thought that this would not be necessary. In future work, this will be an important focus, for both a fully remote system, and a shared one, such as Heimdall.

Remote Control

In addition to the importance of pointing, the instructor would benefit from being able to find the appropriate views without needing to ask the student to do so. This would mean that there is either a recorded interface, such as Heimdall, or for a live remote debugging, the ability for them to remotely control the robotic gantry. This could be done if the system were connected to software and a server as well, but again, we felt that this would be too much for a solution that needed to be ready as quickly as possible.

Conclusion

We believe that this solution would be feasible and easy to use in the short term, but hope to explore the options in our Future Work section and conduct more user tests. Learning remotely is becoming increasingly prevalent so remote debugging and demonstrations are extremely important and urgent. This may mean a much more formalized software and hardware system that will encompass teaching live demos, live debugging, and recorded interactive interfaces for students to explore on their own time. The ideas we explored in this project will definitely serve as a good starting point, but there is still much to be done.