Stroboscope

Design Document

SDMAY19-30
Client: Matt Post
Adviser: Matt Post
Team Members/Roles:
Jessica Bader - Chief Software Engineer / Communication Manager / Scribe
Meghna Chandrasekaran - Chief Software Engineer / Meeting Facilitator
Katrina Choong - Chief Hardware Engineer / Timeline Manager
Seth Noel - Chief Hardware Engineer
Kyle Zelnio - Project Manager
sdmay19-30@iastate.edu
sdmay19-30.sd.ece.iastate.edu

Revised: 10/12/18 - Version 1

Table of Contents

List of figures/tables/symbols/definitions (This should be the similar to the project plan)

1 Introduction

1.1 Acknowledgement

Special thanks to the following people who graciously helped us complete this project in a timely manner and offered their time and resources to us.

Matthew Post - Guidance, Equipment, Financial Aid

Lee Harker - PCB Designing and Printing

1.2 Problem and Project Statement

1.2.1 Problem Statement

EE 448 is a course on AC circuits and motors. In the labs of this course there are mounted motors that the students will do tests with and on. In one of the labs, the students will use a stroboscope to measure the RPM of the mounted motor.

The issues with the stroboscopes that the department has provided for these labs is that they are too delicate for the students using it, the repairs due to its delicacy are too expensive for the department, and it is overcomplicated for the needs of the lab.

1.2.1 Solution Approach

The general solution is to design a substitute stroboscope. This stroboscope will need to be durable enough to be used by students many times, inexpensive to build, repair, and debug, and simple while also meeting the needs of the course and its labs.

The solution of durability will be accomplished by mounting the stroboscope to the motor and encasing the main components. This should reduce the amount of contact the students need to make to the stroboscope and its internals. A more durable stroboscope will reduce the expensive on the department, but not quite enough.

To make an even less expensive stroboscope, parts used for the stroboscope will need to be easily accessible to the department. Parts that the department already use, or from a distributor that is already in use, will make building and repairing inexpensive. A modular design that will allow replacement of smaller portions of the circuit will also reduce the expensive of the stroboscope.

Finally, meeting the needs of the lab is a must. However, making it simple enough for the user to use is also important. By making a GUI simple enough to understand, and making a circuit that will match the RPM of the motor will solve the issue of an over complex stroboscope.

This project is important because the Iowa State Department of Electrical and Computer Engineering works hard to provide high quality education at a minimum cost. This project will decrease the spending of the department, which will lead to lower costs for higher quality for the students in the department.

1.3 Operational Environment

The expected operating environment for the stroboscope would be room 1102 in Coover Hall. It is a lab room for students taking the class EE 448. The stroboscope will operate under room temperatures, meaning there will not be many environmental obstacles. The system should be safe from over heating; however, it is likely that the board and LEDs may become hot from extended use, but they should be safe to touch. The stroboscope will be an enclosed structure and will not be affected by dust.

The stroboscope's software component will be run on the computers found in room 1102 in Coover Hall. It will be accessible to anyone who has a login account to computers found in that room, meaning students and faculty.

If there are any problems with the stroboscope modular model or anything wrong with software components, there will be error detection provided on the software side for users to better understand where the problem roots from.

1.4 Intended Users and Uses

The intended final use of the stroboscope will be in two of the EE 448 labs, to measure the rotational speed of a mounted, rotating disk. The end users will be the students in the course. EE 448 is a course designed for Mechanical Engineering students at Iowa State University, who generally take the course around their sophomore year. EE 448 is a course outside their intended field of study, meaning the students will likely not want to spend large amounts of time in this lab. Therefore, the stroboscope needs to be intuitive to use. We need to expect that students will not be careful with the equipment and take protective measures to make it resistant to breaking.

In addition, the people working for the ETG will be users of our product. They will need to use our documentation to assemble the modules into the final working product. They will also need to use the debugging information to identify problems and repair the stroboscopes when they break. The ETG faculty have a strong background in engineering and fixing lab hardware, which means they will be competent users and will benefit from technical information which can help them debug and potentially improve the stroboscope. However, they are busy and will not be experts in this specific piece of equipment. Therefore, assembling the equipment should be straightforward, clear, and concise so they do not waste any more time than necessary on activities such as assembling the parts and debugging common issues.

Another user who will be using the stroboscope indirectly will be the EE 448 professor. The professor will be grading the labs and will have a set of expectations for the quality of his students' labs. Therefore, the stroboscope must fall within a range of accuracy deemed acceptable by the professor.

1.5 Assumptions and Limitations

Assumptions:

- Will be used by two students at a time, mainly in the EE 448 Motors Lab
- Will be interfaced solely by a Python GUI running on Windows
- Will have an equal brightness to the existing Stroboscope (3300 Lux @ 6000rpm)
- The LEDs will be driven by the Tiva TM4C123G Board
- Final design will be mounted above the motor in the lab and be stationary
- Wall wart powering the LEDs not the Tiva Board
- The lab will not be drastically changed in the near future

Limitations:

- The limit of RPMs should not be below 100 RPM or exceed 2000 RPM (the values used in the lab)
- The size of the stroboscope should be no larger than the motor it is evaluating (as determined by the motors in Coover 1102)
- The circuit should be able to dissipate heat (to avoid breaking)
- The cost to produce the end product should not exceed $500 for each stroboscope (the cost of the previous stroboscope)

1.6 Expected End Product and Deliverables

The end product that will be delivered to the client prior to the end of the project will be a functional, simplified, cost-efficient stroboscope for students taking the course EE 448. We plan to have a partially functioning prototype by December, a fully functional, usable prototype by March to test out with the students in EE 448, and a final product finished by the end of second semester in May. All prototypes will be improvements of the previous version. With the final stroboscope, there will be documentation for the software and hardware side.

- Partially Functioning Prototype - December 2018

    - The first prototype will be used as a basic starting point to show to our client to determine if we have understood all of our requirements for the project properly. It will consist of a simple, functioning GUI that can be used for testing and working LEDs and a Tiva board for the hardware side of the project.
- Fully Functioning Prototype - March 2019
    - The fully functioning prototype will be a prototype that has all the functionality that we would like it to have. This prototype will be used by the students in the EE 448 lab, and we plan to use their feedback for improvements on our final stroboscope product.
- Final Product: Stroboscope - May 2019
    - The final product of the stroboscope will have a fully functioning GUI, which will allow the students in the EE 448 lab to control the stroboscope on a computer; it will be mounted on the motor, so that students can't drop it; it will be more cost efficient and easier to repair; and it will provide error handling tools to help debug when things do not work. The final product will also have the following:
        - Design Documents - The hardware and software side will provide block diagrams.
        - Code - There will be frontend and backend code for both the GUI and the Tiva board being used. The code will be available on GitLab.
        - Software Manual - The software manual will be provided so that anyone who would like to recreate the stroboscope can use our documentation for programming the GUI and Tiva board.
        - Hardware Manual - The hardware manual will be provided so that anyone who would like to recreate the stroboscope has a clear understanding on how to recreate the hardware aspect of the stroboscope.

2. Specifications and Analysis

2.1 Proposed Design

In order to make a stroboscope we first had to think about how we were going to get LEDs to flash. There were three ways we thought we could do this: a Tiva board, an Arduino board, and an AC circuit. Then we realized in any case we would have some circuit to put together. We thought that choosing between a modular design and a fully connected and soldered circuit would be a few good alternatives. We also needed to decide how to case the stroboscope, whether it would be mounted or handheld. Finally, we needed to choose between including a physical knob to adjust the speed on the stroboscope or only allowing interaction with the device to happen through the GUI.

On the hardware side, we have been a little stunted because of waiting for parts, however, we do have a simplified version of a stroboscope working using a bench power supply and some extra parts from ETG to create a mock-up design that could prove our idea in practice while we waited on research and parts to come in from shipping.

With this setup we had an arduino with the default Blink library loaded into it which signaled a MOSFET that was being powered from the bench power supply that was lighting up our extra LED array that we salvaged from a magnetic work light. From testing these lights the bench power supply showed us that we will need at most 2 amps at 5 volts from whichever power supply we end up choosing.

We are working towards a simple PCB design that will allow quick connectors and replaceable parts to be printed then we have to work towards a enclosure that mounts onto the existing motor structure in the Lab.

On the software side, we have started to implement the GUI using Python. In order to create the GUI using Python, we are using Tkinter, which is a Python binding to the Tk GUI toolkit. We have done some testing while trying to implement the GUI; however, the GUI is not yet connected to the functions of the Tiva board that we are programming. In addition to the GUI, we have also started to work on the PWM signal that we need to send to the Tiva board to allow the LEDs to flash. There is no way of testing thing unless we have all the right configurations completed. We have worked on putting together the the files necessary for the set up and plan to move forward from there.

Functional Requirements:

- It will be able to perform all the functions required by the lab
- It will be flexible enough to allow potential future changes to the lab (as described by the course professor)
- It will have accuracy consistent with expectations for the lab
- It will be documented sufficiently to facilitate module compilation
- It will be usable in a manner consistent with the backgrounds of EE 448 students
- Average 3300 Lux @ 6000RPM 12” away from target
- ±5% in RPM fluctuations from original Stroboscope
- Stationary mount
- Wall power supply (No batteries)
- No moving parts
- Modular design for quicker repairs

Non-Functional Requirements:

- It will be more cost-effective to replace than the current version
- It will be replaceable at a module level
- It will be documented sufficiently to allow design improvements to both the software and the hardware
- It will be resistant to breaking due to physical abuse by students

- It will be a user friendly GUI
- It will be repairable by ETG
- It will be cost-effective design compared to old Stroboscope
- It will have easily accessible parts from ETG

2.2 Design Analysis

Our team as a whole has done plenty of researching to determine the best approach for solving the problems of the current stroboscope. From the beginning, we split our team into software and hardware groups, which has helped us with distributing work evenly throughout the team.

On the software side, our goal currently is to get a simple PWM signal programmed to the Tiva board to test the flashing of the LEDs. In order to do so, we have started to create all the files needed for setting up the PWM signal. For this part of the software side, we have not done any testing yet because testing would require all the configurations to be completed. Another functionality on the software side that needs to be completed is the GUI that will be available for the students taking EE 448 on the lab computers in room 1102 in Coover Hall. For the GUI, we are using Tkinter, as mentioned previously. Tkinter is a Python binding to the Tk GUI toolkit. We have done some testing while implementing the GUI, but we have not finished implementing it to fully test it out. Also, the GUI will not be connected to our Tiva board for the first prototype. All functionalities that will be available on the GUI, will be performed by hardcoding the Tiva board for the initial prototype. Eventually, we hope to have some sort of error detection that will be used for testing. The error detection will provide us with compilation errors, and we will then be able to use those error messages to debug our software.

On the hardware side, we tested the LEDs, the transistor, and then the circuit as a whole. We first had access to the transistor for testing. We had researched what we thought would work for to meet the functional requirements it would effect. So we tested it to see if it could switch fast enough and it seemed to be. Then we got the LEDs we ordered. We wanted LEDs that were white, fast switching, and bright enough to match the other stroboscope. Our research seemed good so we tested the LEDs when they came in for those functions individually, and they worked well.

3 Testing and Implementation

3.1 Interface Specifications

The first and main interface between the hardware and the software which will be used in testing is the waveform which will be output by the software to control the hardware. The software will output a square wave which will be input by the hardware as the source of power and connect to the LED. This will cause the LED to flash at the rate of the waveform. The hardware will also have an additional current source which will match to the waveform.

For later testing, we will add in feedback which gets output by the hardware and input by the software. This will be in the form of pins which the hardware sets the voltage to high to indicate a specific situation. The software will watch these pins and have an interruption to handle the problem sent by the hardware.

3.2 Hardware and software

3.2.1 Hardware

During the testing phase of the hardware side. We found good use with the bench power supplies, function generators, and multimeters to accurately create a stroboscope to test the LEDs we had on hand and compare them to the current Stroboscope by brightness and switching speed. Once we found a good brightness we started adding LEDs until we found a good amount of range in brightness at the same voltage level across all of the LEDs. This allowed us to measure the amount of current we are drawing to find a suitable power supply.

The bench power supply allowed huge amounts of current to be drawn from the LEDs at different voltages and read out how much current the supply was delivering.

The function generator allows us to create a square wave at certain frequencies without the need from the software team to give us a working product before we even have working hardware.

The multimeter allowed us to measure the turn-on voltage of the LED array and helped us determine the correct resistor value for our gate on the MOSFET. Without the correct resistance we wouldn't have high enough voltage for the LEDs to turn on from the power supply, or too much voltage causing permanent damage on the LEDs.

3.2.2 Software

When testing, the extent to which software will be used in testing will be the compilation software. This will notify us when we have compilation errors and we will be able to use print statements in the software to debug where the errors occur. This will be a useful way to detect small errors which may stop the software from running (such as misspellings or incorrectly used variables) as well as segmentation faults.

We will also use the output waveform to test the hardware. We will output a waveform at variable speeds to ensure the circuit is behaving as expected and blinking the LED at the correct speed. It will also help us determine if the circuit can successfully function at its minimum and maximum speeds. This will be helpful because once we have created a waveform, we will know what our output is and therefore what expected result should happen from the circuit.

3.3 Functional Testing

We will be using the process of software testing to make sure that the required functions are met for the stroboscope on the software side of of our project. We will also be using a testing process with the general idea of the software testing process for the entirety of the project. In the end it will:

- be able to perform all the functions required by the lab
- be consistently accurate with expectations for the lab
- be documented sufficiently to facilitate module compilation
- be usable in a manner consistent with the backgrounds of EE 448 students

3.4 Non-Functional Testing

Testing for performance, security, usability, compatibility

We plan to frequently test and monitor performance and compatibility through our design process and multiple revisions.

Our stroboscope will be:

- a user friendly GUI
- resistant to breaking due to physical abuse by students
- replaceable at a modular level documented sufficiently to allow design improvements to both the software and hardware
- satisfactorily accurate to perform the expectations of the lab

3.5 Process

So far we have only tested the transistor and are working on testing the LEDs and the PWM mode. To unit test the transistor we looked into what functional requirements were affected by it, and then tested it based on those requirements. The main requirement was switching speed. So we made a simple circuit to test on it at a current and voltage level we thought was plenty enough and but a bench waveform through it to test the switching speed.

3.6 Results

When testing the transistor for functional requirements we were pretty successful. We concluded from the testing that the transistor we had chosen was satisfactory for our needs. We simulated our circuit using a bench waveform generator.

4 Closing Material

4.1 Conclusion

Our main goal is to create a stroboscope that is mounted, modular, and user friendly to withstand over time, have more cost effective repairs, and provide easier functionality for students in the lab.

By splitting into software and hardware teams, the project is more manageable and allows each member to contribute their strengths to the overall project. The software team focuses on how to interface with the user. While the hardware team designs the circuit which will have the electrical components to make the stroboscope function. We plan to create multiple prototypes to test the integration of hardware and software components to make sure the test stroboscope has met the requirements of the client.

So far, the software team has created a GUI and is continuing to refine how to take user input and create a waveform using PWM to send a frequency so the LEDs will flash. The hardware team has created a test model to simulate potential brightness, finding peak voltage, and overall design of how the LEDs will work on future prototypes. We are also working together to get a PCB design with the help of Lee Harker in ETG integrated into our design process.

Once the software and hardware teams have solidified their parts are fully functional on their own, we will combine the two together as a whole system. By creating multiple prototypes, we will test to make sure the system receives input from the software side to produce an output on the hardware side and resolve any issues.

4.2 References

This will likely be different than in project plan, since these will be technical references versus related work / market survey references. Do professional citation style(ex. IEEE).

4.3 Appendices

Any additional information that would be helpful to the evaluation of your design document.

If you have any large graphs, tables, or similar that does not directly pertain to the problem but helps support it, include that here. This would also be a good area to include hardware/software manuals used. May include CAD files, circuit schematics, layout etc. PCB testing issues etc. Software bugs etc.