

Stroboscope

PROJECT PLAN

SDMay19-30

Client: Matt Post

Advisers: Matt Post

Jessica Bader - Chief Software Engineer, Scribe, Communication
Manager

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

<http://sdmay19-30.sd.ece.iastate.edu>

Revised: 9/28/18/Version 1

Table of Contents

1	Introductory Material	1
1.1	Acknowledgement	1
1.2	Problem Statement	1
1.3	Operating Environment	2
1.4	Intended Users and Intended Uses	2
1.5	Assumptions and Limitations	3
1.6	Expected End Product and Other Deliverables	3
2	Proposed Approach and Statement of Work	5
2.1	Objective of the Task	5
2.2	Functional Requirements	5
2.3	Constraints Considerations	5
2.4	Previous Work And Literature	5
2.5	Proposed Design	7
2.6	Technology Considerations	11
2.7	Safety Considerations	12
2.8	Task Approach	12
2.9	Possible Risks And Risk Management	13
2.10	Project Proposed Milestones and Evaluation Criteria	14
2.11	Project Tracking Procedures	14
2.12	Expected Results and Validation	15
2.13	Test Plan	17
3	Project Timeline, Estimated Resources, and Challenges	20
3.1	Project Timeline	20
3.2	Feasibility Assessment	22
3.3	Personnel Effort Requirements	22
3.4	Other Resource Requirements	23
3.5	Financial Requirements	24
4	Closure Materials	25
4.1	Conclusion	25

List of Figures

Figure 2.4.1: Current Stroboscope

Figure 2.4.2: DIY Stroboscope

Figure 2.5.1: High Level Stroboscope Design

Figure 2.5.2: High Level Software Design

Figure 2.5.3: High Level Hardware Design

Figure 2.8.1: Project Approach

List of Tables

Table 2.5.1: Interface Pro/Con

Table 2.5.2: Circuit Pro/Con

Table 2.5.3: Board Pro/Con

Table 2.12.1: Requirements and Corresponding Tests

Table 3.1.1: Project Timeline

Table 3.3.1: Time Estimation of Tasks

List of Symbols

List of Definitions

AC: Alternating Current

ETG: Electronics Technology Group

GUI: Graphical User Interface

LED: Light Emitting Diode

PCB: Printed Circuit Board

RPM: Rotations Per Minute

1 Introductory Material

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 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 using a stroboscope. A stroboscope measures the rotational speed of an object by flashing a light on an object. The object has a single small image in one place on the circle; when the object spins, if the light flashes at exactly the speed it takes the object to make one rotation, then the image will appear stationary. Using this information, the stroboscope is able to calculate the rotational speed of the object.

In addition, these stroboscopes are too delicate for the students using it, the repairs due to its delicacy are an unnecessary expense for the department. In addition, the stroboscope being used is overcomplicated for the needs of the lab. It has a range from 30-300,000 RPM with an accuracy of 0.005%. In comparison, the lab only makes use of measurements from 100-2000 RPM and other pieces of lab equipment have errors which are much greater than this. It also has other features which are not used, such as allowing an external input instead of the flashing of the stroboscope itself to calculate the speed. These are features which are being paid for but are not being used.

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. It will match the current stroboscope in terms of attributes which are used (such as lumen intensity) but decrease the aspects which are not being used in the lab (such as a much larger range than required, multiple input options, etc.).

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. The students will also interact with the stroboscope only through the GUI. By decreasing student interaction with hardware, we will decrease how often the stroboscope is exposed to harsh conditions. 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.

1.3 OPERATING 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., EE 442, EE452 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 INTENDED 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 motor shaft. 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. The lab, room 1102 in Coover Hall, is locked at all hours, so the students would have to make sure they complete their lab during their allotted lab hours. Therefore, the stroboscope needs to be intuitive to use, which is why our GUI will eliminate time for setting up the stroboscope. 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 if they break. The ETG Staff 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 specifications deemed appropriate 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 OTHER 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 - The code will follow IEEE coding standards. 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 Proposed Approach and Statement of Work

2.1 OBJECTIVE OF THE TASK

The objective of our project is to design a stroboscope for students to use in the EE 448 lab. This stroboscope will be more cost efficient, more specifically designed for the needs of the lab, easier to use, easier to repair, and less breakable. It will consist of both hardware and software components. In order to accomplish these goals, we plan to make our new stroboscope a modular design so that repairing will be easy; we plan to use the Tiva board rather than an Arduino because the Tiva board is more accessible and cheaper to replace if anything were to go wrong; we plan to make it mounted so it's less likely that students could damage it; and we plan to use a GUI rather than a knob on the hardware so that it's much easier to use and less breakable.

2.2 FUNCTIONAL REQUIREMENTS

- It will be able to measure the rotational speed of a mounted, rotating motor shaft as per the needs of the lab
- It will be flexible enough to allow potential future changes to the lab with the modular design and the use of the Tiva boards
- It will have an accuracy ranging between $\pm 1\%$ to $\pm 5\%$
- It will be documented sufficiently to facilitate module compilation
- It will be easier to use considering the background of the EE 448 students with a simple GUI for user interaction

2.3 CONSTRAINTS CONSIDERATIONS

- 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

2.4 PREVIOUS WORK AND LITERATURE

Other stroboscopes that are currently on the market do the same thing in different ways. The main thing that a stroboscope will do is strobe light at some consistent rate. The point is to direct this strobing light at a rotating object so that the object appears stationary. The stroboscope needed by the lab we are designing this stroboscope to be used in, must be able to display how fast the stroboscope is flashing so that the user can determine the rotational speed of the motors used in this lab.

It is important to note that the previous stroboscope used by this lab was on the cheaper side of the market. When purchasing a new stroboscope kit [directly from Monarch](#)

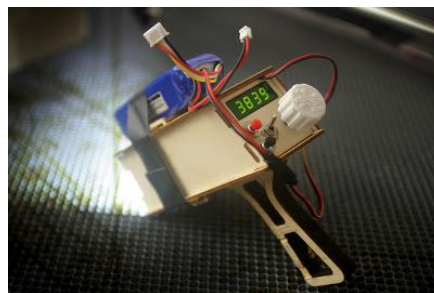
[Instruments, the manufacturer](#), the cost is \$499 for one stroboscope kit (Figure 2.4.1). A different stroboscope of very similar quality and function, though it lacks in some important areas, is priced at \$1399.99 new [from the manufacturer](#), Fluke Electronics. Another stroboscope found on the market, which is missing functions or is not as rigorous in its functions, is \$695 new [from the manufacturer](#), Shimpo Instruments. Still another stroboscope, from Testo Instruments, priced at \$886.75 [on Newark](#), and this stroboscope has similar but lacking functions. These prices of course only reflect the first time purchase, and not the upkeep and repair of the stroboscopes, which is another factor that the stroboscope we are designing will be taking into account.



*Figure 2.4.1. Current Stroboscope
Monarch Instrument*

Taking many things into perspective for this project, such as price to produce (and reproduce), price to repair, time to repair, functional requirements, and more, we will design a stroboscope that meets the needs of the lab for the class EE 448. That is, though it is a stroboscope, it is a stroboscope with the EE 448 lab as a focus, unlike other stroboscopes on the market, which have the general public as a focus. As a second focus area, we want the ETG to be able do the producing, reproducing, and repair easily and cost effectively.

Before starting any design work on our project we inspected DIY designs of stroboscopes across the internet (Figure 2.4.2) and are taking notes on the science and functionality behind flashing LEDs.



*Figure 2.4.2. DIY Stroboscope
William Osman*

2.5 PROPOSED DESIGN

In order to make a stroboscope we first had to think about how the stroboscope would be controlled; would be a hardware-only circuit controlled with a knob, a software-controlled circuit controlled by a GUI, or a software controlled circuit with the additional option of a knob? We created the following list of pros and cons:

Table 2.5.1: Interface Pro/Con

Knob		GUI		Both	
Pros	Cons	Pros	Cons	Pros	Cons
<ul style="list-style-type: none"> - Quick to adjust - Simplifies the design 	<ul style="list-style-type: none"> - Not using software limits control - Knobs are breakable - Our group has more software knowledge than hardware - Tuning would be less accurate 	<ul style="list-style-type: none"> - Breaks less often - Makes work division in our group easier due to interests - Can develop faster - Can adjust on a finer level 	<ul style="list-style-type: none"> - Relying on USB connection 	<ul style="list-style-type: none"> - Software can still provide control - Gives the user options 	<ul style="list-style-type: none"> - More complex - Already limited on time - Knob is still breakable

After discussing with our client, we decided that the most important factor was breakability; our client wanted to limit damage which would require repair to the tool because students in labs are rough on equipment. This led us to decide on the GUI only design. The software-only interface would limit how often students broke the stroboscope.

Next, we needed to decide 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. We created the following list of pros and cons:

Table 2.5.2: Circuit Pro/Con

Tiva board		Arduino		AC circuit	
Pros	Cons	Pros	Cons	Pros	Cons
<ul style="list-style-type: none"> - Built-in clock - Can use software - Cheaper for ISU - Already available - Many features - More ISU resources 	<ul style="list-style-type: none"> - No experience in the group - Bad IDE - Difficult 	<ul style="list-style-type: none"> - More tutorials online - Nice IDE - Multiple board sizes - Past experience in group 	<ul style="list-style-type: none"> - Less GPIO than Tiva - Would need to order - Expensive for ISU 	<ul style="list-style-type: none"> - Experience - Less components 	<ul style="list-style-type: none"> - Handling frequency changes - Not configurable with computer - Harder to debug

After discussing with our client, we decided that the most important factors were low cost and ability to have a software interface. This meant we decided to use the Tiva board, which was available to the department at a low cost because they were already buying it in bulk and would provide an easy software interface.

After speaking with our client, we were brainstorming ways to facilitate repair of the tool. We came up with the idea to create a modular design of interconnected pieces which would allow replacement of individual modules in the case of damage. We created the following pro/con list:

Table 2.5.3: Board Pro/Con

Modular Board		Singular Board	
Pros	Cons	Pros	Cons
<ul style="list-style-type: none"> - Facilitates repair 	<ul style="list-style-type: none"> - Breakable at the 	<ul style="list-style-type: none"> - Easier repairs 	<ul style="list-style-type: none"> - Any part breaking

- Decreases cost of repair	joints - Requires more space - Someone would need to debug the problem (time to repair increases, along with human cost)	(replacement only) - Faster repairs - More compact - Cleaner design	would require full replacement - More expensive in terms of materials
----------------------------	--	--	--

After discussing with our client, we found he liked the idea of being able to replace individual parts and did not foresee the human time to be too costly. We decided to design a modular board.

After making these design decisions, we determined that our design would include two components at the highest level; a hardware component and a software component. The hardware component would flash the LEDs while the software component would interact with the user, adjust the speed of the flashing, and provide the power which would facilitate the LED flashing. Their main interface will be software to hardware. However, we hope to add some hardware to software feedback to facilitate software-controlled prevention from scenarios which could cause damage (such as overheating) and debugging which component needs to be replaced when one breaks.

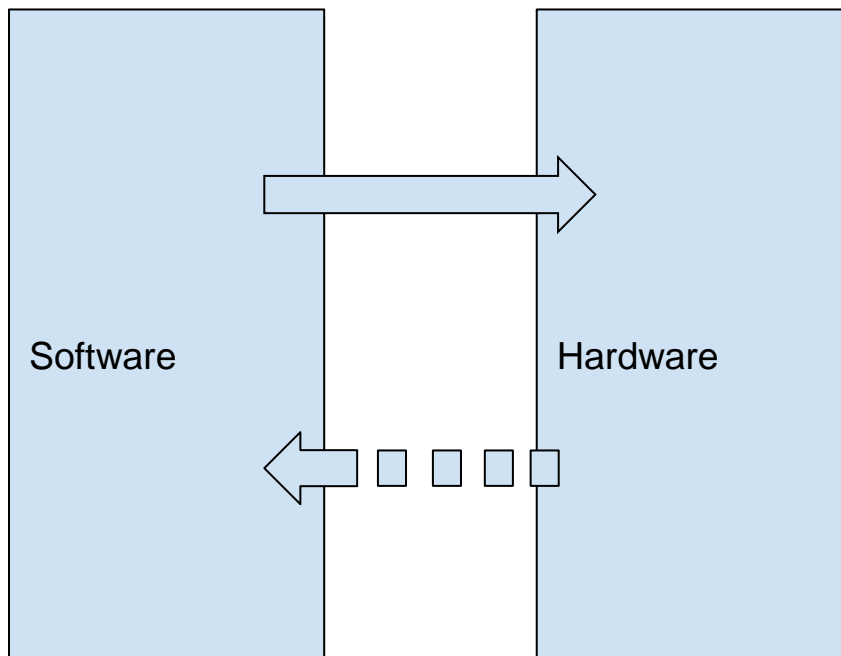


Figure 2.5.1. High level stroboscope design

The software would be broken into two components; a GUI and the microcontroller. The GUI would take input from the user and communicate to the microcontroller. The microcontroller would be in charge of using this information and the feedback from the hardware to change the speed of the LED flashing, then calculating updated information to return to the GUI.

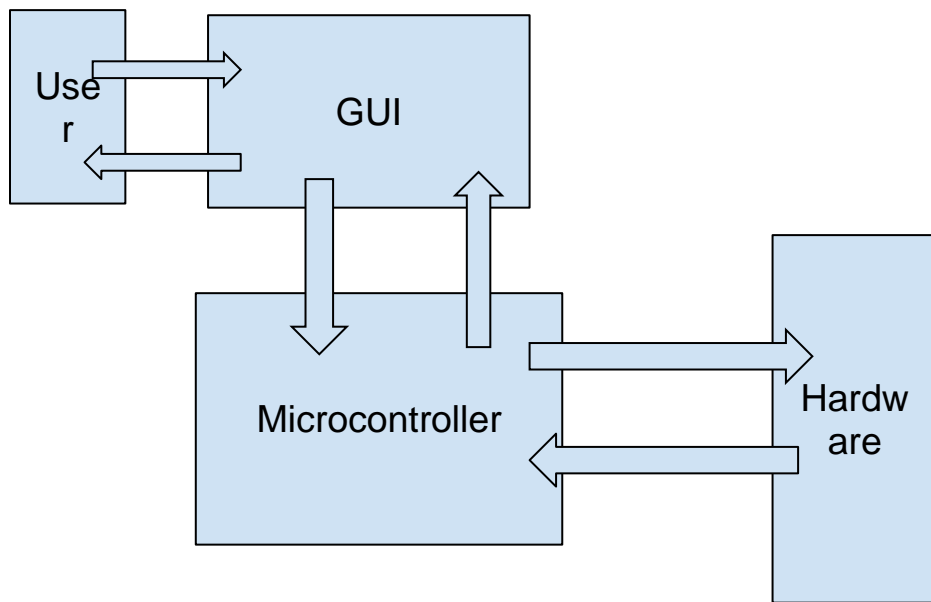
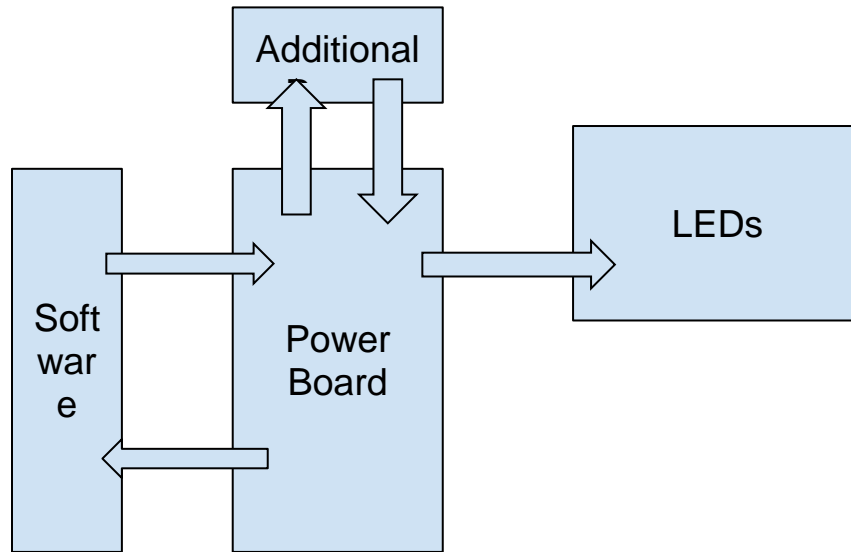


Figure 2.5.2. High level software design

Finally, the hardware would take the output from the software, pass it through a power board to control how it is used, and into a set of LEDs which would blink. It will also use an additional current supply, because the Tiva board does not output enough current.



The power board would return the necessary information to the software.

Figure 2.5.3. High level hardware design

2.6 TECHNOLOGY CONSIDERATIONS

Before making any definite decisions on what types of technology we wanted to use, we created a pros and cons list for all our potential approaches.

We were debating using either a Tiva board or an Arduino for the hardware necessary for the stroboscope components. If our stroboscope project was to be recreated, choosing the Arduino would make the total cost of the stroboscope more expensive compared to using the Tiva board. Also, the Arduino has less GPIO pins, which gives us less input pins. The Tiva board is much cheaper, easily accessible, has more features, has more GPIO pins, and more resources at ISU. Because of these reasons, we decided to use the Tiva board instead of an Arduino.

When deciding on whether to use AC power or the Tiva board, we concluded that using AC power would be more tedious to control the change in frequency. Using AC power would require more electrical components meaning the system would be more expensive, more complex, less reliable, and harder to debug issues throughout the process. It would also not be configurable with a computer, which is our main goal to make our stroboscope

user friendly via GUI. On the other hand, the Tiva board has an onboard clock that can be used for flashing the LEDs, and it is configurable with software.

Since one of the major issues with the current stroboscope used by students in the lab for EE 448 is the lack of care when using the item, our goal is to use as little user friendly components to the mounted stroboscope. Therefore, when debating whether or not to add a knob on the mounted stroboscope to increase and decrease speed, we decided on not using a knob because that's another piece that would need to be repaired and could easily be damaged. Also, the accuracy when changing speed with the knob would not be as precise. Instead of the knob, we decided all the user functionalities will be done on the software side to keep parts from being damaged. The GUI we are creating will take care of handling speed change for the stroboscope and will be more accurate.

When determining languages for the software side, we decided on using Python to create the GUI. Our decision on Python was based on the fact that some of our team members have experience using Python. Also, our client, Matt Post, provided us with previous Python files for a GUI that we are using as examples to help get started. For programming the Tiva board, we are using C to configure all our needs to the Tiva board.

2.7 SAFETY CONSIDERATIONS

As with any electrical device, it will be important to not touch any exposed wires to prevent electrical shock. In addition, use should be terminated immediately if the stroboscope becomes hot or releases smoke; these could be signs of insufficient heat release and could lead to fire, explosion, and/or damage to the product. The team will uphold all standards of IEE code conduct to ensure the safety of the users, and the creators follow the safety policies.

2.8 TASK APPROACH

Describe any possible methods and/or solutions for approaching the project at hand. You may want to include diagrams such as flowcharts to, block diagrams, or other types to visualize these concepts.

We decided to use a design thinking approach in which we utilize a series of prototypes which are tested and improved upon. Our cycle will begin with identifying the problem. This will include defining what did/did not work and where we can make improvements. We will also take client feedback at this stage. Next, we will brainstorm ideas to improve our product, evaluate these ideas, and discuss them with our client. After this stage, we will prototype our new design. Finally, we will test our product to evaluate its performance. At this point, we will also evaluate whether or not our client and end users are happy with our product; if they are we are finished, if not then we will start again at the first stage. Below is a detailed diagram of our task approach:

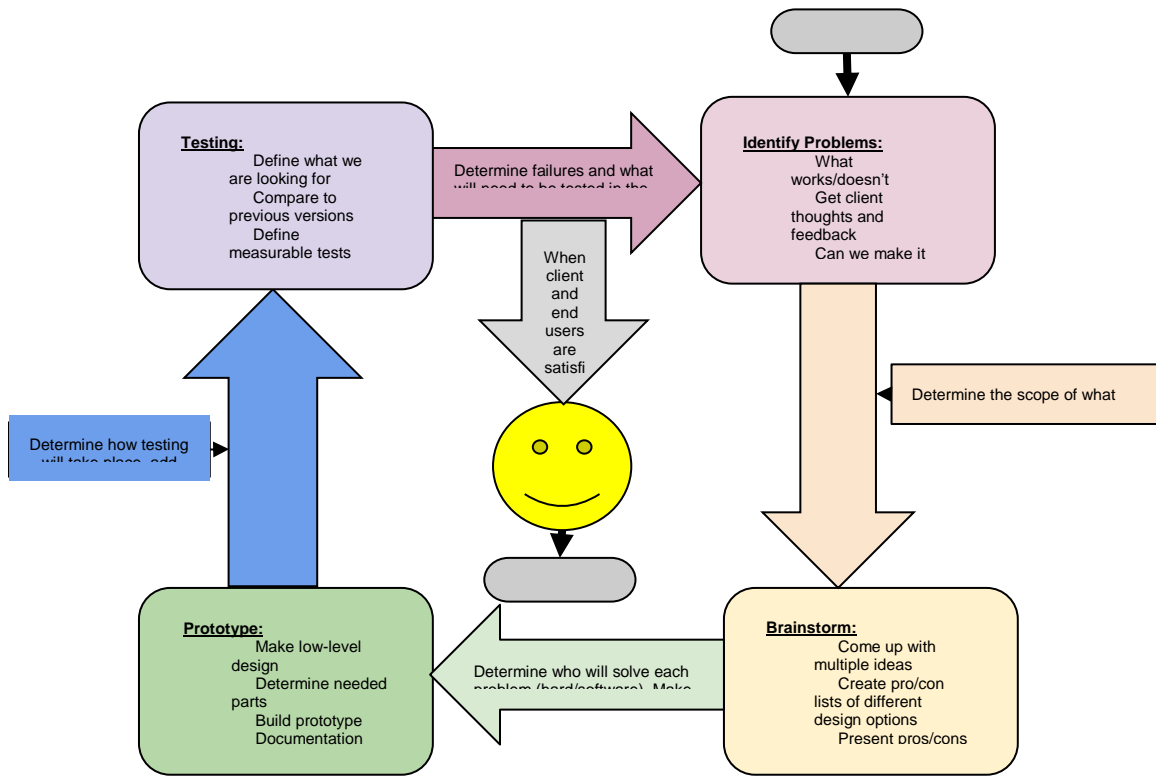


Figure 2.8.1. Project Approach

2.9 POSSIBLE RISKS AND RISK MANAGEMENT

Looking at our current rough design of what we want, it will be somewhat of a hurdle to get the specific SMD LEDs online as the shipping time will slow things down. Also, when we get to designing and printing the PCBs waiting for those to be sent might halt our progress until those arrive.

Currently, no one in our group has any experience in designing circuit boards and we see this as a huge learning step to overcome and are seeking help from Lee Harper to aid us in getting our first design completed and printed.

Using the Tiva board, we are unsure of how accurate our clock speed will be and the latency of our circuit will add to the accuracy of the stroboscopes RPM measurement. We are trying for $\pm 1\%$ to $\pm 5\%$ accuracy for the expected RPM from our stroboscope to the RPM of the current stroboscope.

2.10 PROJECT PROPOSED MILESTONES AND EVALUATION CRITERIA

The key milestones will be prototypes one through four. Prototype one will flash the LEDs from the output of the Tiva board controlled by the GUI. This output will not be complicated or very controlled. The second prototype will do the same but now with some determined accuracy. It is to be +/-25% accuracy at this point. Prototype three and four are to be determined. March it is expected to be functional enough to test in the labs.

2.11 PROJECT TRACKING PROCEDURES

Our group will use weekly status reports for each member to keep track of our team's progress throughout the course of the two semesters. The weekly reports will contain detailed information of what we have completed during that week, how many hours we worked during that week, and what we plan to accomplish in the next week. Each team member will be assigned with a task for the week and should aim to complete those tasks in a timely manner. However, since problems may come up, the deadlines are tentative. As a team, we will hold each other accountable for the work we are responsible for to make sure people don't fall behind or contribute less as a whole compared to other members.

Progress Tracking Measures

Prototypes - We will use our prototypes as a way of tracking our project progress by making sure we achieve all our goals for the planned prototypes, such as the partially functioning prototype (completed by December 2018) and the fully functioning prototype (completed by March 2019).

Weekly meetings with client - During our weekly meetings with our client, we explain what progress we have made on the project. The software team and hardware team explain their accomplishments, problems they faced, and what they plan to work on for the coming week.

Weekly meetings with whole team - During our whole group meetings, we discuss each person's progress on their part of the project. Generally, since we have a software and hardware team, each team discusses what they accomplished and what they are looking for from the other team to move forward on their part.

Weekly meetings with software/hardware group - During our individual software/hardware team meetings, we work together as a team to overcome any complications an individual team member has faced during the week. This way, we get through problems faster and can move forward, keeping us from falling behind. We also discuss individual progress for the week for software/hardware.

2.12 EXPECTED RESULTS AND VALIDATION

We expect our design to result in a stroboscope which will be able to be used to fulfill the needs of the lab, will be able to be repaired without replacing the whole product, and will be less expensive to replace than the current stroboscope. We will know our product is sufficient when we have verified it can be used to perform the lab, we have tested that the product can be assembled based on the documentation provided, we have tested that students can successfully use the stroboscope based on our documentation, and our client is content with the end product. The tests we will use to verify these criteria:

Table 2.12.1: Requirements and corresponding tests

Requirement	Test(s)
Accuracy of 1% from 100 to 2,000 RPM	Rotate the motor at 100 RPM and measure the speed with our stroboscope. Compare this with the same measurement on the original stroboscope (which has an error of 0.005%). Ensure our measurement is within 1% of the original stroboscope's.

	Repeat the above test at 2,000 RPM.
	Repeat the above test at 10 more evenly distributed speeds.
Fulfill the needs of the lab	Test the stroboscope in the lab scenario by going through the lab. Ensure with the grading professor that the results we measure would receive full credit for the lab.
Can be repaired	Replace each replaceable component on the stroboscope. Repeat the measurement test for 100 RPM, 2,000 RPM, and one speed between.
	Give our documentation to an ETG worker and ensure they can successfully build the product.
Useable by the students	Allow the students in the course to use the stroboscope in lab. Ensure they can complete the steps required in the lab without external assistance (excluding the documentation designed for facilitating student understanding of the tool).

2.13 TEST PLAN

Based on our functional requirements, we will conduct the following tests to determine that the product has successfully fulfilled our requirements:

Functional Requirement #1: It will be able to to measure the rotational speed of a mounted, rotating motor shaft as per the needs of the lab

Test Case:

For this requirement, we want to make sure the stroboscope is accurate within the required range of 100 to 2,000 RPM with an accuracy between 1% to 5%.

Test Steps:

1. Rotate the motor at 100 RPM
2. Measure the speed with our stroboscope
3. Measure the speed with the original stroboscope
4. Determine if the measurement is within 1% to 5% of the original stroboscope
5. Repeat the first 3 steps for 2,000 RPM

Expected Results:

The rotational speed we measure using our stroboscope should be within 1% to 5% accurate in comparison to the rotational speed measured from the original stroboscope.

Functional Requirement #2: It will be flexible enough to allow potential future changes to the lab with the modular design and the use of the Tiva boards

Test Case:

For this requirement, we want to test the stroboscope in scenarios where different modules could be replaced and determine how easy it is to replace them.

Test Steps:

1. Reprogram another Tiva board with the functionalities of the original Tiva board and determine if there are any complications with this step
2. Reconnect the GUI to the new Tiva board and determine any complications with this step
3. Take apart the modules (so the PCB and the Tiva board) and determine how much time it takes to put it back together
4. Have ETG Staff recreate our stroboscope using our documentation to determine if it well written to the point that recreation of the stroboscope is fast and easy

Expected Results:

We expect the replacement and repairing of the stroboscope to be easy and fast because it only has two modules and since it uses the Tiva board, it should be easily accessible to the ETG Staff to get ahold of a new Tiva board to reprogram if anything were to go wrong.

Functional Requirement #3: It will have an accuracy ranging between $\pm 1\%$ to $\pm 5\%$

Test Case:

Similar to the first functional requirement, for this requirement, we want to make sure the stroboscope is accurate within the required range of 100 to 2,000 RPM with an accuracy between 1% to 5%.

Test Steps:

1. Rotate the motor at 100 RPM
2. Measure the speed with our stroboscope
3. Measure the speed with the original stroboscope
4. Determine if the measurement is within 1% to 5% of the original stroboscope
5. Repeat the first 3 steps for 2,000 RPM

Expected Results:

The rotational speed we measure using our stroboscope should be within 1% to 5% accurate in comparison to the rotational speed measured from the original stroboscope.

Functional Requirement #4: It will be documented sufficiently to facilitate module compilation

Test Case:

For this requirement, we want to make sure our documentation is well written so that our project can be recreated by the ETG Staff for all the lab stations in the EE 448 lab.

Test Steps:

1. Re-read our documentation and make sure most (if not all) uncertainties are clarified
2. Have ETG Staff recreate our stroboscope using our documentation to determine if it well written to the point that recreation of the stroboscope is fast and easy with little to no uncertainties

Expected Results:

Considering the ETG Staff background and their technical skills, the ETG Staff should be able to recreate our stroboscope in less than half the time it took us to create the stroboscope since all documentation should be clear and precise on specifications and building instructions.

Functional Requirement #5: It will be easier to use considering the background of the EE 448 students with a simple GUI for user interaction

Test Case:

For this requirement, we want to test the stroboscope in the lab scenario by going through a normal lab for EE 448.

Test Steps:

1. Mount our stroboscope to the motor in the EE 448 lab
2. Open the GUI on the lab monitors
3. Input an RPM into the GUI anywhere between 100 to 2,000 RPM
4. Make sure everything is set up properly
5. Redo the above 4 steps a couple times

Expected Results:

The stroboscope we create should be much easier for the students to use because the simple GUI makes input and functionality much easier. Also, the fact that the stroboscope will be mounted should mean there is a less likely chance the students could break the device. We expect the device to work smoothly and perform similar to the original stroboscope.

3 Project Timeline, Estimated Resources, and Challenges

3.1 PROJECT TIMELINE

Our overall process is to use an Agile-like method with a series of prototypes which we test and make adjustments to accordingly. Our timeline includes due dates for each prototype, as well as what we are expecting specifically from the early prototypes. We are also hoping to test the product in lab by March.

Stroboscope

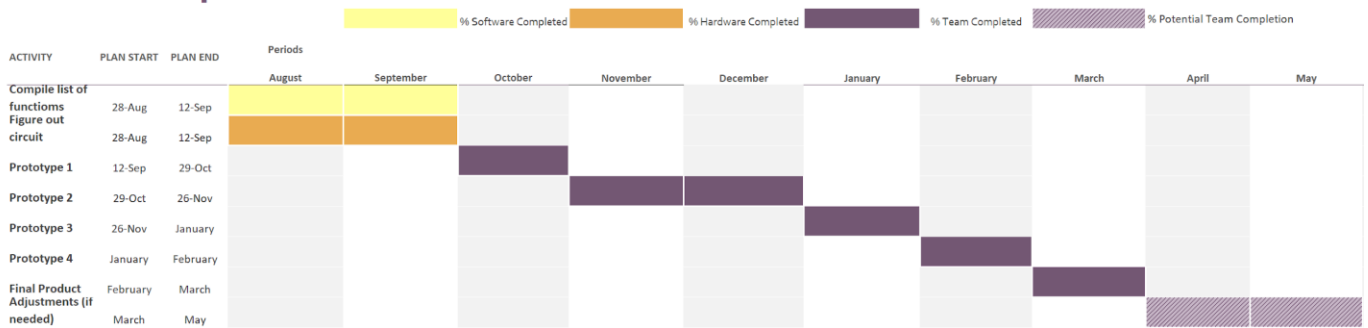


Table 3.1.1: Project Timeline

Semester 1:

Date	Software	Hardware
August-September 2018	<ul style="list-style-type: none"> • Compile a list of functions needed for the software 	<ul style="list-style-type: none"> • Figure out circuit components and values needed:
October 29, 2018	<ul style="list-style-type: none"> • Prototype Stage 1: Can create an output wave and have a GUI which can take an input and affect the output wave. 	<ul style="list-style-type: none"> • Prototype Stage 1: Circuit carries out basic functions; taking input from software and outputting a flash
November 26, 2018	<ul style="list-style-type: none"> • Prototype Stage 2: GUI can take an input and change the output wave within 25% precision, and calculate the RPM within 25%. 	<ul style="list-style-type: none"> • Prototype Stage 2: Have a wide accuracy range, 25-30% to output flash; Have first PCB prototype

Semester 2:

Date	Software	Hardware
January	<ul style="list-style-type: none"> • Prototype Stage 3: 	<ul style="list-style-type: none"> • Prototype Stage 3:
February	<ul style="list-style-type: none"> • Prototype Stage 4: 	<ul style="list-style-type: none"> • Prototype Stage 4:
March	<ul style="list-style-type: none"> • Have finished product completed • Test in EE448 lab with students 	<ul style="list-style-type: none"> • Have finished product completed • Test in EE 448 lab with students
March-May	<ul style="list-style-type: none"> • Consider received feedback from client and students, make adjustments accordingly 	<ul style="list-style-type: none"> • Consider received feedback from client and students, make adjustments accordingly

3.2 FEASIBILITY ASSESSMENT

The stroboscope will consist of a mounted enclosure with an array of LEDs pointing towards the motor's shaft and be bright enough to match the current stroboscope's viewing capabilities.

Creating simple and reliable circuits to keep costs down and repairs to a minimum will be challenging as we would want a working prototype of the stroboscope by January 1st 2019.

We are hoping that the time between circuit board printing does not hinder the progress of our project and we find ways around the foreseeable delay of shipping times.

3.3 PERSONNEL EFFORT REQUIREMENTS

First, we will need to do the planning of our project, including tasks like writing requirements and working with our client. The software team will need to create a GUI and a variable waveform. The hardware team will need to design the basic circuit which will include the microprocessor through the LED. Next, we will start testing the product including systems testing of how the hardware and software work together. The entire product needs to be heavily documented for the specific purposes of our client. Later in the project, we will integrate protection for the hardware and feedback which will help ETG engineers debug to fix broken stroboscopes. One of the last parts will be to design the case for the stroboscope.

Time 3.3.1: Time Estimation of Tasks

Task	Description	Hours
Project Planning	Before we can start working, we need to plan our approach. This includes tasks such as defining customer needs, defining requirements, creating a timeline, doing high level designs, and defining interfaces.	30
GUI	Implement a GUI which can be used to command the stroboscope, tell the user the speed, facilitate debugging, and give the user a simple explanation.	80
Waveform	The software must be able to create and output a waveform which will control the blinking of the LED. This needs to be adjustable across the range of speeds for the stroboscope and accurate to the specifications of the stroboscope.	80
Circuit design	We need to design a circuit which will be external to the computer and will include the microcontroller, the LED, a second current supply,	100

	and any other needed circuitry, along with connections between components.	
Testing	We will need to test the product from a system level, to see if the hardware and software portions work together in the way we are expecting and that the final product fulfills the requirements.	50
Documentation	We need to create detailed documentation to explain our designs and purposes. We also need to create the needed documentation to help users at all levels operate the stroboscope and update the product in the future if needed.	60
Case design	We will need to design and create the case which the stroboscope will be encased in.	20
Protection	We must include hardware features which protect the stroboscope from errors which may break it such as surge protection, heat dissipation, and preventing the stroboscope from going too fast.	80
Feedback	We need to implement circuitry and software to take feedback and facilitate basic debugging if the product breaks.	80

3.4 OTHER RESOURCE REQUIREMENTS

The other resources required for the software side include an IDE when working on the GUI code in Python, an IDE for writing code onto the Tiva board, and a computer to run the GUI on for students in EE 448 to use. The IDE that we are using for creating a GUI in Python is PyCharm by JetBrains. PyCharm provides code inspections and error highlighting, which will help when trying to debug. The IDE that we are using for writing code onto the Tiva board is Code Composer Studio. This will allow us to easily write code and transfer it to the Tiva board. We will be using the computers in room 1102 in Coover to run the GUI.

The other resources required for the hardware side include purchasing parts from DigiKey, getting parts from the ETG shop in Coover Hall, using frequency analyzers and function generators, and software to create a PCB design. The parts that we need ordered from DigiKey will be purchased by our client, Matt Post. The parts needed from the ETG shop are accessible during the open hours of the shop. Frequency analyzers and function generators are provided to us for use in the Senior Design Lab located in Coover Hall. In

order to create the PCB design, we plan to a software called EAGLE, which allows us to create the design and print it out.

3.5 FINANCIAL REQUIREMENTS

The predicted financial requirements for this project will all be hardware-based costs corresponding directly to the price of the components.

One-time fees will include:

- Electrical components used to build prototypes (we foresee four to five prototypes before the final design and one extra design for additional feedback from client)
 - Digikey purchases
 - ETG parts

Recurring fees will include:

- Price of all the hardware pieces required to build the final design (including any printed boards or 3D printed casing)
- Replacement parts (a subset of the above pieces, but not always a full set)
- Manpower to assemble the pieces based on documentation
- Upon breakage of the product, manpower to debug the problem and replace broken components

4 Closure Materials

4.1 CONCLUSION

The current stroboscopes used by students in EE 448 are frequently breaking due to mistreatment and usage over time. With that, there is constant need for repairs causing more output expenses. We plan to create a mounted, modular, and user friendly stroboscope to withstand over time, more cost effective repairs, and providing 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.

As a team, we plan to take the necessary steps as outlined in this documentation to merge software and hardware components seamlessly to produce a functional stroboscope that fits all the requirements of the client's needs.

4.2 REFERENCES

[#] Monarch Instrument, "PLS Pocket LED Stroboscope," *Monarch Instrument*. [Online]. Available: <https://monarchinstrument.com/products/pls-pocket-led-stroboscope?variant=31280174792>. [Accessed: Oct. 28, 2018].

T. Bigelow. "EE 448 Lab 6 Report.doc." EE 448, Iowa State University. Ames, Iowa.

T. Bigelow. "EE 448 Lab 5 Report.doc." EE 448, Iowa State University. Ames, Iowa.

"High-Power LED Stroboscope." *electronicsforu.com*, 22 June, 2015.

<https://electronicsforu.com/electronics-projects/hardware-diy/high-power-led-stroboscope>

LargeMouthBass, "Stop Time With an LED Stroboscope!" *Instructables*, <https://www.instructables.com/id/Stop-Time-with-an-LED-Stroboscope/>

Monarch Instruments 2018, *Pocket LED Strobe Instruction Manual*, instruction manual, Monarch Instruments, U.S.A.

M. Post.

Texas Instruments 2007-2014, *Tiva TM4C123GH6PM Microcontroller Data Sheet*, data sheet, Texas Instruments Incorporated, Austin, TX.

W. Osman. "Stroboscope." 3 Oct. 2016.

<http://www.williamosman.com/2016/10/stroboscope.html>

4.3 Appendices

If you have any large graphs, tables, or similar that does not directly pertain to the problem but helps support it, include that here. You may also include your Gantt chart over here.

- Any additional information that would be helpful to the evaluation of the project plan or should be a part of the project record shall be included in the form of appendices
- Examples of project documentation that might be included are property plat layouts or microprocessor specification sheets germane to the proposed project.