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: 12/3/18/Version 3

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 6

2.5 Proposed Design 8

2.6 Technology Considerations 13

2.7 Safety Considerations 14

2.8 Task Approach 14

2.9 Possible Risks And Risk Management 14

2.10 Project Proposed Milestones and Evaluation Criteria 15

2.11 Project Tracking Procedures 15

2.12 Expected Results and Validation 16

2.13 Test Plan 17

3 Project Timeline, Estimated Resources, and Challenges 24

3.1 Project Timeline 24

3.2 Feasibility Assessment 24

3.3 Personnel Effort Requirements 24

3.4 Other Resource Requirements 26

3.5 Financial Requirements 26

4 Closure Materials 28

4.1 Conclusion 28

4.2 References 29
4.3 Appendices 31

List of Figures

Figure 2.4.1: Current Stroboscope

Figure 2.4.2: DIY Stroboscope

Figure 2.5.1: Pulse

Figure 2.5.2: High Level System Design

Figure 2.5.3: Capacitor Drain Graph 1

Figure 2.5.4: Capacitor Drain Graph 2

Appendix A: Circuit Diagram

Appendix B: Process Diagram

Appendix C: Gantt Chart (semester 1)

Appendix D: Gantt Chart (semester 2)

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 3.3.1: Time Estimation of Tasks

Table 3.4.1: One-time Fees

Table 3.4.2: Recurring Fees

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 have graciously helped us complete this project in a timely manner and offered their time and resources to us throughout the first semester of our project.

Matthew Post - Guidance, Equipment, Financial Aid

Lee Harker - PCB Designing and Printing

Timothy Bigelow - Project Feedback

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. The stroboscope being used is also overcomplicated for the needs of the lab. It has a range from 30-300,000 RPM with an accuracy of 0.005% [1]. 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.

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 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 overheating; 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 perform all the functions required by the lab
- It will be able to have 0.5% accuracy
- It will range from 100 to 3000 RPM
- Average 3300 Lux at 6000 RPM, 12" away from target
- It will have a GUI for user interaction

2.3 Constraints Considerations

Constraints and non-functional requirements of the project include:

- It will be more cost-effective to replace than the current version
- It will be user friendly in a manner consistent with the backgrounds of EE 448 students
- It will be documented sufficiently
- It will be flexible enough to allow potential future changes to the lab
- It will be resistant to breaking due to physical abuse by students
- It will have easily accessible parts for the ETG

In this project, there are several IEEE standards which we will be following to ensure our project is high quality, ethical, and comparable to other products:

1. IEEE Standard for Software and System Test Documentation [2]
   This standard creates a process by which to acquire data, create documentation, and maintain documentation. Our product is to be used in a lab for years after we have left ISU. Therefore, the documentation of our product is essential for the maintainability of the product. By following this standard, we will ensure that our documentation will allow future users at all levels to successfully use the product. We will also ensure that upgrades and maintenance can be made on the system so it will be usable over a long period, even if the lab needs are slightly modified.

2. IEEE Standard for System, Software, and Hardware Verification and Validation [3]
   This standard defines the verification and validation process used to review a product to determine whether it satisfies the requirements and the user's needs. We will use this follow this standard to verify that our system meets the requirements that we have defined. By using this method to prove we have satisfied our requirements, we will be able to evaluate whether we have met our requirements and prove to our client this is the case.

3. IEEE Guide for Selecting and Using Reliability Predictions Based on IEEE 1413 [4]
   This standard outlines factors which can be used to determine the reliability of a product, defines what information should be included in the reliability report, and identifies acceptable ways to collect data regarding product reliability. One of our most important functional requirements is that the stroboscope can operate with a 0.5% reliability from 100 RPM to 3000 RPM. Therefore, we will need to evaluate how reliable our stroboscope is across this range. We will use this standard to determine reliability factors, compile information for our reliability report, and develop a method to collect data. Following this standard will best evaluate whether our product meets this criterion and will give substance to our final report for our client.

4. IEEE Recommended Practices for Modulating Current in High-Brightness LEDs for Mitigating Health Risks for viewers [5]
   This standard discusses modulation frequencies which pose a potential risk to users who suffer from epileptic seizures. It recommends practices which can be used to minimize these risks. Because our stroboscope will flash LEDs, any students taking the course who suffer from epileptic seizures could be put at risk. While we cannot completely eliminate the strobing, understanding tactics to minimize these risks will be essential to the health of some users. Our product should be inclusive and as safe as possible for all users.

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). According to its specifications, it can measure with a 0.005% accuracy from 30 to 300,000

RPM[6]. A different stroboscope of very similar quality and function is priced at $1399.99 new from the manufacturer, Fluke Electronics. This stroboscope can also measure from 30 RPM to 300,000 RPM, however it only has a 0.02% accuracy level. We notice that it is more expensive for less accuracy[7]. Another stroboscope found on the market is $695 new from the manufacturer, Shimpo Instruments. This stroboscope has a lower range of 60 RPM to 120,000 RPM and an accuracy of 0.01% [8]. Again, it is less accurate, has a lower range, and is more expensive than the current stroboscope being used. Still another stroboscope, from Testo Instruments, priced at $886.75 on Newark. This stroboscope has the same specifications as the stroboscope from Fluke Electronics, with a range from 30 RPM to 300,000 RPM and an accuracy of 0.02% [9]. 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. Therefore, we can see from this data that the stroboscope currently being used is the least expensive stroboscope on the market, we the best specifications for that price. However, the EE 448 lab only needs to go from 234 RPM to 1804 RPM due to the limitations of the speed of the motor used in lab [10],[11]. Therefore, there is no option on the market for which the ETG does not have to pay for more than they are using.

Figure 2.4.1. Current Stroboscope
Monarch Instrument [1]

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. We have a description on *Instructables* which gives us a high-level explanation of how a stroboscope works. It also includes a video which shows how a stroboscope looks when it has found the correct speed [13]. We also found example stroboscope designs to understand the technical implementation of a stroboscope. One

example is the stroboscope design we found at *electronicsforu* which included a PCB design along with an explanation of how a stroboscope works [14]. We were able to use this as a starting place for our stroboscope.

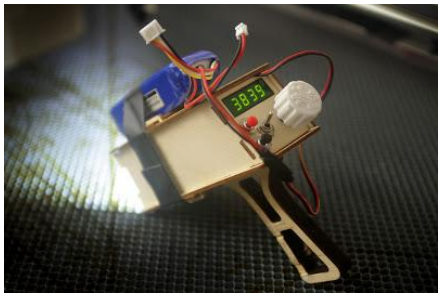


Figure 2.4.2. DIY Stroboscope
William Osman [12]

2.5 Proposed Design

In order to make a stroboscope we first had to think about how the stroboscope would be controlled; would it 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
- Quick to adjust - Simplifies the design	- Not using software limits control - Knobs are breakable - Our group has more software knowledge than hardware - Tuning would be less accurate	- Breaks less often - Makes work division in our group easier due to interests - Can develop faster - Can adjust on a finer level	- Relying on USB connection	- Software can still provide control - Gives the user options	- 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 lead 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
- Built-in clock - Can use software - Cheaper for ISU - Already available - Many features - More ISU resources	- No experience in the group - Bad IDE - Difficult	- More tutorials online - Nice IDE - Multiple board sizes - Past experience in group	- Less GPIO than Tiva - Would need to order - Expensive for ISU	- Experience - Less components	- 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
- Facilitates repair - Decreases cost of repair	- Breakable at the joints - Requires more space - Someone would need to debug the problem (time to repair increases, along with human cost)	- Easier repairs (replacement only) - Faster repairs - More compact - Cleaner design	- Any part breaking 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 forsee the human time to be too costly. We decided to design a modular board.

At the heart of our design lies the microcontroller. This is what controls the strobe effect. The microcontroller will output a pulse wave (as shown in Figure 2.1.2) with a low value around 0V and a high value around 3V. When the pulse is high, the LED will turn on and when the pulse is low the LED will turn off. Therefore, the period of the pulse wave will determine the speed at which the stroboscope flashes.

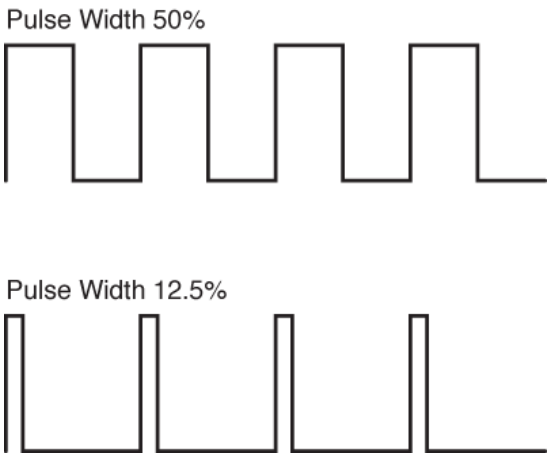


Figure 2.1.5 [15]

We have determined that our design will include two components at the highest level; a hardware component (shown in red in Figure 2.1.1) and a software component (shown in blue in Figure 2.1.1). During our design and test process, these two parts will be

implemented and tested on their own before integration and testing of the entire system. Furthermore, the software side can be divided into to modules, because we are using different software languages to program the GUI and the microcontroller.

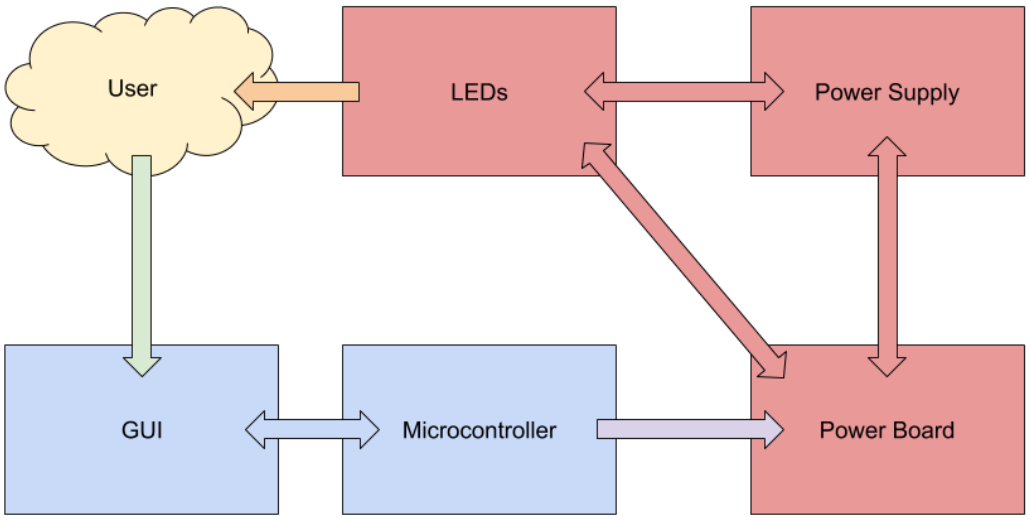


Figure 2.5.2

Module 1: The GUI

The first portion of the software component is the GUI. The user will be able to modify the pulse length through the GUI. They will have two options; they will be able to go to a specific speed (for example, 500RPM) or they will be able to adjust up or down 1-5RPM from their current speed. This will allow the user to start with a guess as to the speed of the motor, then adjust it by small amounts when they get close. The GUI will be programmed in Python using Tkinter, which is a Python binding to the Tk GUI toolkit. However, the Tiva board will be programmed in C. This means we will need to use a pipe to communicate between the GUI and the microcontroller. The GUI will be able to send a function call to the Tiva board and retrieve the output of that function.

Module 2: The Microcontroller

The microcontroller also belongs to the software component. The waveform it outputs will be modifiable. We have determined that hardware will need to be designed for a specific rise period (this will be better explained when discussing the hardware), so the rise period of the wave will always be consistent, regardless of the speed of the wave. We have identified that a 100μs rise period will be ideal with our current LEDs. To come to this conclusion, we have resources from our research which describe that the rise period needs to be as short as possible to maximize the strobing effect. However, the shorter the rise period, the less bright the LED will be. By testing our stroboscope at increasingly short

wave period, we have determined that the strobe effect is clear until 100µs, leading us to determine that this is the optimal rise length.

We will use the microcontroller to output the wave by using PWM mode on the microcontroller. The Tiva board operates at 80Mhz and has 24 bits which can be used to create the period of the output wave [16]. 24 bits allows for the microcontroller to count as high as od16777215.

$$16{,}777{,}215\ units\ /\ 80{,}000{,}000\ (units\ /\ s) = 0.2097\ s$$

$$1/0.2087\ s = 5RPS$$

$$5RPS * 60s = 300RPM$$

Therefore, the built-in PWM mode of the Tiva board would only allow us to go as low as 300 RPM, when we need to go to 100 RPM. We will solve this by creating our own version of PWM mode using the clock on the microcontroller which will allow us to go from 100 RPM to 300 RPM.

The Tiva board will output this wave on the pin PB5. This pin, as well as the ground pin, will be taken as inputs to the hardware circuit.

Module 3: The Hardware Circuit

Our hardware is designed to 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 will blink. It will also use an additional power supply, because the Tiva board does not output enough current to make the LEDs bright. The circuit board will return the necessary information to the software. A diagram of the circuit design is referred to Appendix A. Circuit Schematic.

The LEDs have a turn on and turn off time that we must observe because there will be a delay between the software output waveform to the hardware. To counter the delay, we have measured the voltage output of the LEDs and measured the rise time which is our Δx at 4.6µs (Figure 2.1.3) and a fall time of 3.6µs (Figure 2.1.4). In total we have a rise and fall period of 8.2µs which can be translated to the microcontroller to allow for the LEDs to have peak brightness and create a proper strobing effect.

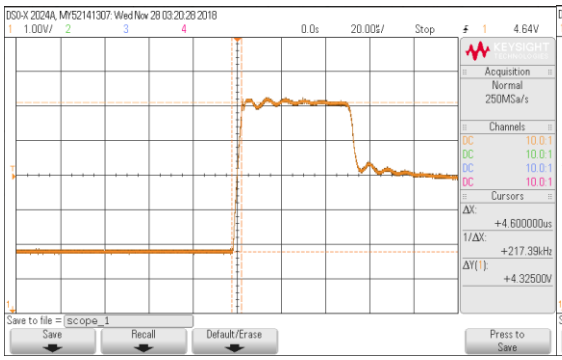


Figure 2.5.3

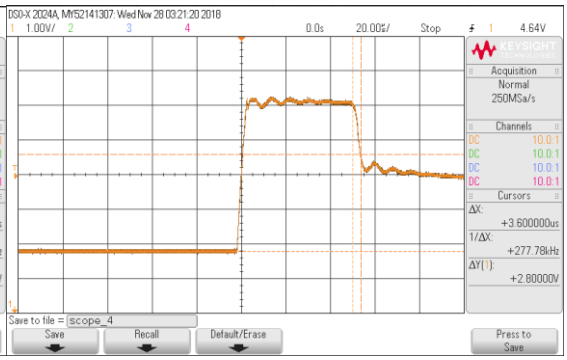


Figure 2.5.4

We have added an NMOS to act as a switch in our circuit to receive the signal from the Tiva board to switch on and off the LEDs. Adding capacitors has increased the brightness by raising the peak voltage of our LEDs while maintaining the pulse width. However with the addition of capacitors, our output waveform has been slightly deformed from the original square wave because of the time it takes for the capacitors to drain (see Figure 3.1.1).

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 IEEE code conduct to ensure the safety of the users, and the creators follow the safety policies.

2.8 Task Approach

We decided to use a design thinking approach in which we utilize a series of prototypes which are tested and improved upon each iteration. Our cycle will begin with identifying the problem shown in the top right corner of the Process Diagram (Identify Problems, Appendix B). 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 stroboscope, evaluate these ideas, and discuss them with our client. Using these ideas we are going to discuss each pro and con we can think of between our interfaces (Brainstorm, Appendix B). After this stage, we will prototype our new design (Prototype, Appendix B). In this stage, we will make low-level designs to help determine what parts we need. We will also write documentation for the prototypes. Finally, we will test our product to evaluate its performance (Testing, Appendix B). 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 after determining the success and failures at that point. See *Appendix B* for a detailed diagram of our task 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 Harker 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 0.5% accuracy from 100 to 3000 RPM.

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 perform all the functions required by the lab

Test Case:

Based on [10, 11] we were able to determine that the lab will require students to measure the speed of a motor rotating at speeds ranging from 234 RPM to 1804 RPM. Based on these values, we were able to determine that a range from 100 RPM to 3000 RPM will fulfill all needs of the lab.

Test Steps:

1. Complete prototype 5 for the stroboscope
2. Test out this prototype in the EE 448 Lab by allowing the students and TAs to use it
3. Get feedback from the students and TAs
4. Use their feedback to make improvements to our prototype 5 design

Expected Results:

We expect our stroboscope design to fit most, if not all, the needs of the lab after our prototype 5 design. The feedback we will be getting will hopefully be based on simple technical issues.

Current Results:

As of 12/2/18, we have not conducted this test yet. We plan to do it towards the end of next semester once we have finished our 5th prototype.

Functional Requirement #2 & #3: It will have an accuracy of 0.5%, and it will range from 100 to 3000 RPM.

Test Case:

For this requirement, we want to make sure the stroboscope is accurate within the required range of 100 to 3,000 RPM with an accuracy within 0.5%

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 0.5% of the original strobe because it has 0.005% accuracy
5. Repeat the first 4 steps at increments of 200 RPM up to 3,000 RPM

Expected Results:

The rotational speed we measure using our stroboscope should be 0.5% accurate in comparison to the rotational speed measured from the original stroboscope.

Current Results:

As of 12/2/18, all the the tests we have conducted on the stroboscope show that it can range from 300 to 3000 RPM. We have not tested the accuracy of it yet, but we plan to determine the accuracy and make changes to allow it to reach 100 RPM at the beginning of next semester.

Functional Requirement #4: Average 3300 Lux at 6000 RPM, 12 inches away from target.

Test Case:

For this requirement, we want to test the light intensity of the stroboscope when it is 12 inches away from the target and make changes if it does not have an average of 33000 Lux at 6000 RPM.

Test Steps:

1. Shine the stroboscope 12 inches away from a moving motor at 6000 RPM
2. Use a lux meter to determine the light intensity of the stroboscope at that distance
3. Check if the value is around 3300 Lux
4. Make changes if this value is not 3300 Lux at 6000 RPM

Expected Results:

We expect the stroboscope to have a light intensity of 3300 Lux at 6000 RPM when it is 12 inches away from the target.

Current Results:

As of 12/2/18, we have not done this test yet, but we plan to test this in the next semester to make sure that we are getting the right light intensity for our stroboscope.

Functional Requirement #5: It will have a GUI for user interaction.

Test Case:

For this requirement, we want to test and make sure the GUI works.

Test Steps:

1. Open the GUI on the lab monitors
2. Input an RPM into the GUI anywhere between 100 to 3,000 RPM
3. Make sure everything is set up properly
4. Check whether the input RPM is actually changing the values on the Tiva Board
5. Redo the above 4 steps a couple times

Expected Results:

We expect the GUI for the stroboscope to be functional. It should allow users to input RPM values and directly change the values that the Tiva Board will be outputting.

Current Results:

As of 12/2/18, we have a GUI that a user can interact with. However, as of this date, it does not connect to the Tiva Board and change the output values of it yet. We plan to get this functionality -- the GUI changing the output values of the Tiva Board -- to work at the beginning of the next semester.

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

Non-Functional Requirement #1: It will be more cost-effective to replace than the current stroboscope.

Test Case:

For this requirement, we want to determine that the stroboscope is less expensive than the current stroboscope being used

Test Steps:

1. Look up the price of the current stroboscope
2. Total the price to fully outfit a station with a stroboscope

Expected Results:

We expect that our final price to outfit a lab station with a new stroboscope will be less than 25% of the price of the current stroboscope.

Current Results:

At this point, our stroboscope costs $45 in total. This is 9% of the cost of the original stroboscope, so we are currently passing this requirement.

Non-Functional Requirement #2: It will have a user-friendly GUI

Test Case:

For this requirement, we want to determine that the GUI is easily usable with the background level of our student users.

Test Steps:

1. Install our fully-functioning stroboscopes in Coover 1102 before the stroboscope lab for EE 448 in March of 2019
2. Have the students complete the lab
3. Have the students fill out a survey about their experience with the stroboscope, including how they would rate the difficulty on a scale of 0-5, a short-answer explanation of any problems they encountered, and any suggestions they have for improving the usability of the stroboscope

Expected Results:

We expect that we will receive an average rating of less than 20% difficulty. We also expect that we will receive feedback on difficulties which we can use to further improve user-friendliness of the stroboscope.

Current Results:

We have not done this test yet. It will be completed in March, during the spring semester course.

Non-Functional Requirement #3: It will be documented sufficiently.

Test Case:

For this requirement, we want to determine that the documentation of the stroboscope is sufficient

Test Steps:

1. Ask a member of the ETG to assemble our stroboscope from the individual pieces using our instructions
2. Have the ETG member complete a survey which rates the difficulty of the process from 0-5 and asks them to give an explanation of any difficulties they had
3. Have the students complete the lab using the stroboscope use instructions
4. Have the students complete a survey which includes a short-answer description of any problems they had
5. Have our advisor review our hardware design and software code

Expected Results:

We expect that the ETG member and student's will rate each set of instructions with less than 20% difficulty level. We also expect them to have minimal difficulties and give enough feedback for us to diagnose and resolve any issues with the clarity of the instructions. Finally, we expect our advisor to give us the feedback that we have documented our hardware and software well enough for an outsider to understand the design.

Current Results:

We have not conducted this test yet. We will run it in March during the spring semester course.

Non-Functional Requirement #4: It will be flexible enough to allow potential future changes to the lab

Test Case:

For this requirement, we want to determine that we have programmed our software to allow for future updates

Test Steps:

1. All values will be stored in variables (nothing will be hard coded)
2. All code will be commented to explain its function
3. A review will take place at the end to ensure the first two steps have been completed

Expected Results:

We expect that upon review, our code will follow these standards 100% of the time.

Current Results:

A code review on 12/2/18 has determined we are following these standards 100% of the time.

Non-Functional Requirement #5: It will be resistant to breaking due to physical abuse by the students.

Test Case:

For this requirement, we want to determine that the stroboscope has a low level of maintenance required due to the stroboscope breaking.

Test Steps:

1. Install our fully-functioning stroboscopes in Coover 1102 before the stroboscope lab for EE 448 in March of 2019
2. Have the students complete the lab
3. Take note of all requests for maintenance of the stroboscope to the ETG staff
4. Have the Teaching Assistant fill out a survey about their experience with the stroboscope, including if any of the stroboscopes broke or required maintenance of any kind

Expected Results:

We expect that no stroboscopes will require repair during the first semester. We also expect that if one breaks, we will be able to collect enough data through

records from the ETG and TA in order to diagnose and determine a long-term solution to ensuring the stroboscopes do not regularly break in this way.

Current Results:

We have not run this test yet. It will be run in March, during the spring semester course.

Non-Functional Requirement #6: It will have easily accessible parts for the ETG

Test Case:

For this requirement, we want to determine that the ETG will be able to order all parts for the stroboscope easily and through the processes they are already using

Test Steps:

1. Our first choice for parts shall always be parts which the ETG is already ordering for another course or purpose
2. If there are no parts already available through the ETG, we will choose parts from the suppliers the ETG most often uses
3. We shall only order parts from new places when the first two criteria cannot be met
4. A review will take place at the end to ensure we followed these criteria

Expected Results:

We expect that upon review, we will find we have followed these steps 100% of the time when choosing components for our project.

Current Results:

A review of our current parts as of 12/2/18 has determined that we are following these standards 100% of the time.

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. Based on the feedback we receive, we will make adjustments accordingly to demo as our final project. Our Gantt charts can be found in *Appendix C* and *Appendix D*.

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.

Table 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, and any other needed circuitry, along with connections between components.	100
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:

Table 3.4.1: One-time Fees

Part	Price / Unit	Num. Units	Total
Electrical components used to build prototypes (Digikey purchases and ETG parts)	$40	5	$200
Cases for all lab stations	$5	8	$40
Circuit board for all lab stations	$40	8	$320
			$560

Recurring fees will include (as needed):

Table 3.4.2: Recurring Fees

Part	Price / Unit	Num. Units	Total
Single case replacement	$5	1	$5
Single circuit replacement	$40	1	$40
Upon replacement, manpower to assemble the pieces based on documentation	10 mins	1	10 mins of ETG time
Upon breakage, manpower to debug the problem and replace the broken components	1 hour	1	1 hour of ETG time

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

[1] Monarch Instruments, *Pocket LED Strobe Instruction Manual*, Monarch Instruments, U.S.A., 2018.

[2] *IEEE Standard for Software and System Test*, IEEE Standard 829, 2008.

[3] *IEEE Standard for System, Software, and Hardware Verification and Validation*, IEEE Standard 1012, 2012.

[4] *IEEE Guide for Selecting and Using Reliability Predictions Based on IEEE 1413*, IEEE Standard 1413.1, 2002.

[5] *IEEE Recommended Practices for Modulating Current in High-Brightness LEDs for Mitigating Health Risks to Viewers*, IEEE Standard 1789, 2015.

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

[7] Fluke, "Fluke 820-2 Stroboscope, 7 LED Array, 4,800 Lux @ 6,000 FPM/30cm," *Fluke Corporation*, Fluke 820-2. [Online]. Available: https://www.fluke-direct.com/product/fluke-820-2-stroboscope. [Accessed: Oct. 25, 2018].

[8] Shimpo Instruments, "DT-326B High Performance LED Stroboscope," *Shimpo Instruments*, DT-326B. [Online]. Available: http://shimpoinstruments.com/stroboscopes/dt-326b. [Accessed: Oct. 25, 2018].

[9] Newark, "TESTO 477 - LED STROBOSCOPE, 30 TO 300000FPM, 0.02% ROHS COMPLIANT: YES," *newark.com*. [Online]. Available: https://www.newark.com/testo/testo-477/led-stroboscope-30-to-300000fpm/dp/05AC9583?st=strobocope#anchorTechnicalDOCS. [Accessed: Oct. 25, 2018].

[10] T. Bigelow. "EE 448 Lab 5 Report.doc." Unpublished manuscript, EE 448: Introduction to AC Circuits and Motors, Iowa State University, Ames, Iowa, U.S.A.

[11] T. Bigelow. "EE 448 Lab 6 Report.doc." Unpublished manuscript, EE 448: Introduction to AC Circuits and Motors, Iowa State University, Ames, Iowa, United States.

[12] W. Osman, "Stroboscope," Oct. 3, 2016. [Online]. Available: http://www.williamosman.com/2016/10/stroboscope.html. [Accessed: Nov. 28, 2018].

[13] LargeMouthBass, "Stop Time With an LED Stroboscope!" *Instructables*, Nov. 26, 2015. [Online]. Available: https://www.instructables.com/id/Stop-Time-with-an-LED-Stroboscope/. [Accessed: Oct. 28, 2018].

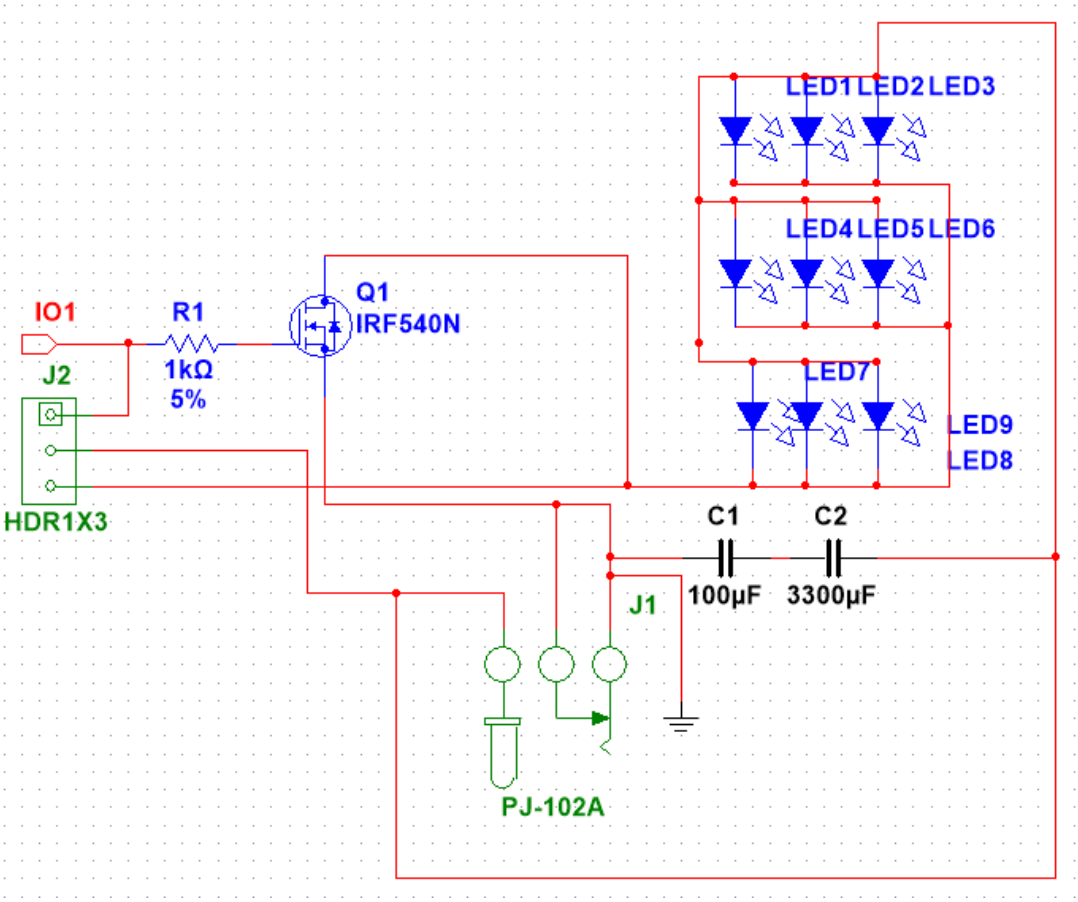
[14] "High-Power LED Stroboscope." *electronicsforu.com,* 22 June, 2015. [Online]. Available: https://electronicsforu.com/electronics-projects/hardware-diy/high-power-led-stroboscope. [Accessed: Oct. 28, 2018].

[15] "Pulse Wave with a Y-Offset," *facer.com*, Oct. 2017. [Online]. Available: http://community.facer.io/t/pulse-wave-with-a-y-offset/11852. [Accessed: Dec. 1, 2018].

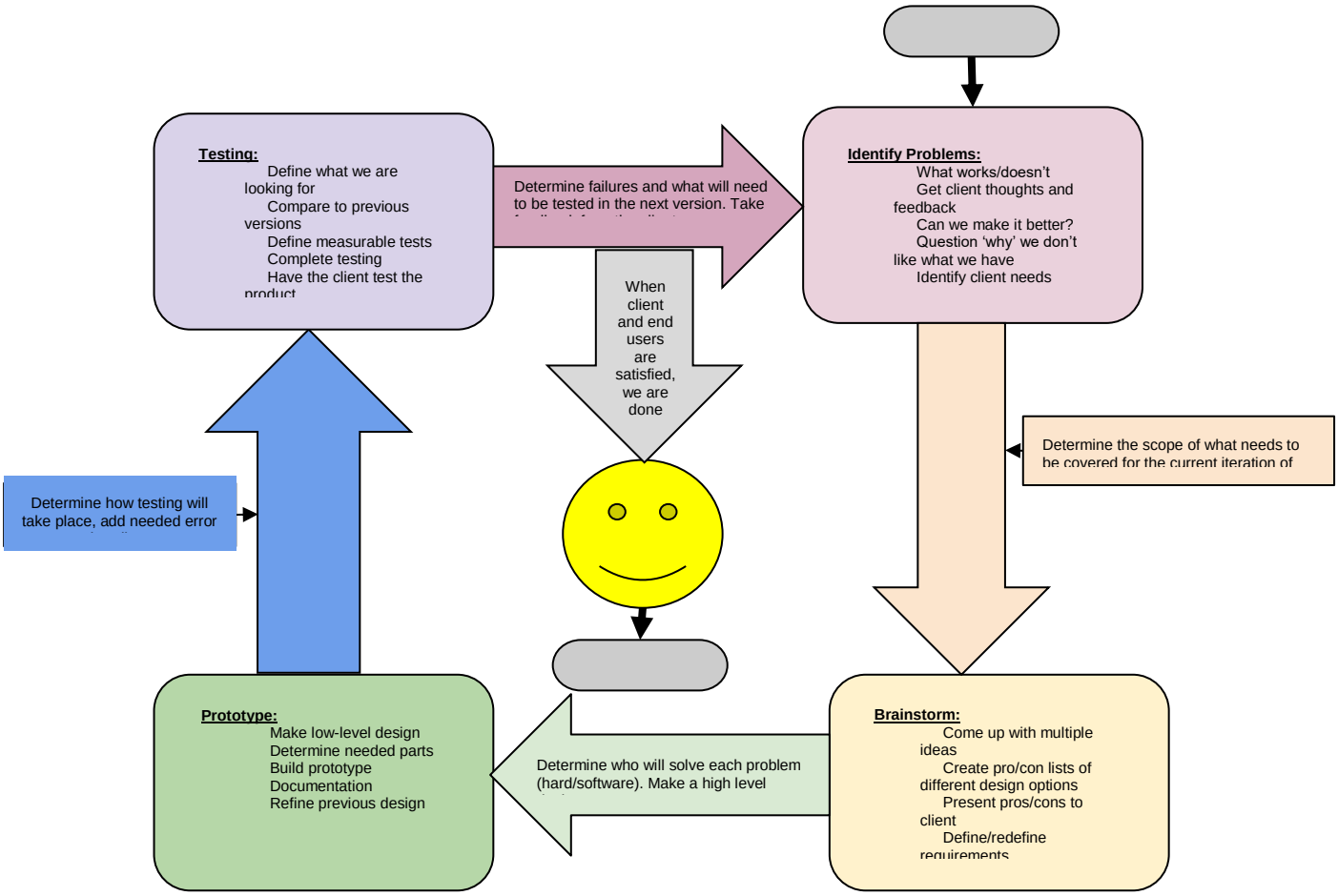
[16] Texas Instruments, "Tiva TM4C123GH6PM Microcontroller Data Sheet," TM4C123GH6PM datasheet, Texas Instruments Incorporated, Austin, TX, U.S.A., June 12, 2014.

4.3 APPENDICES

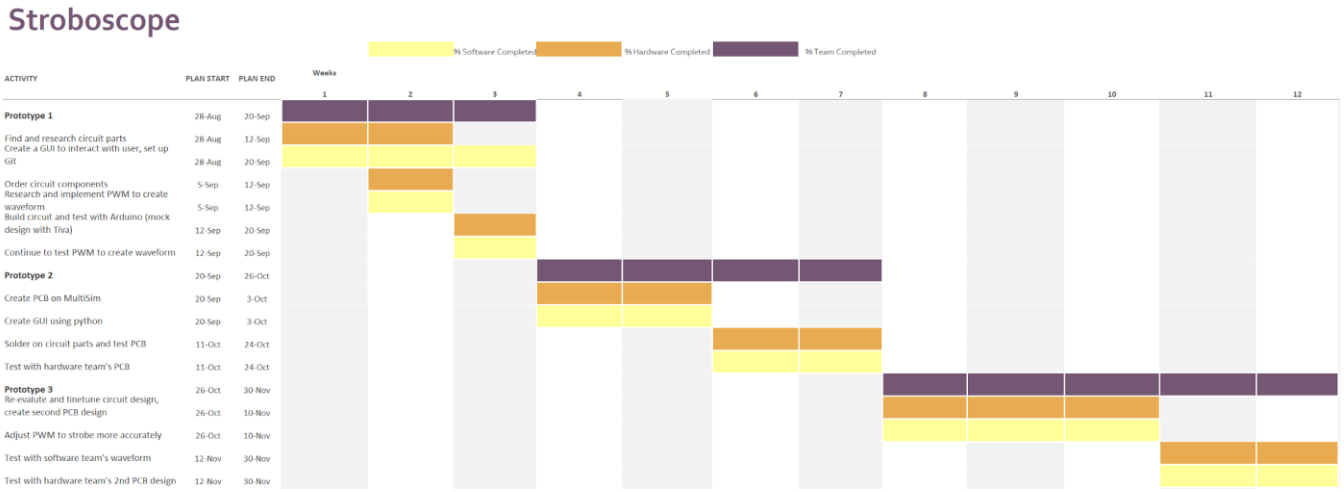
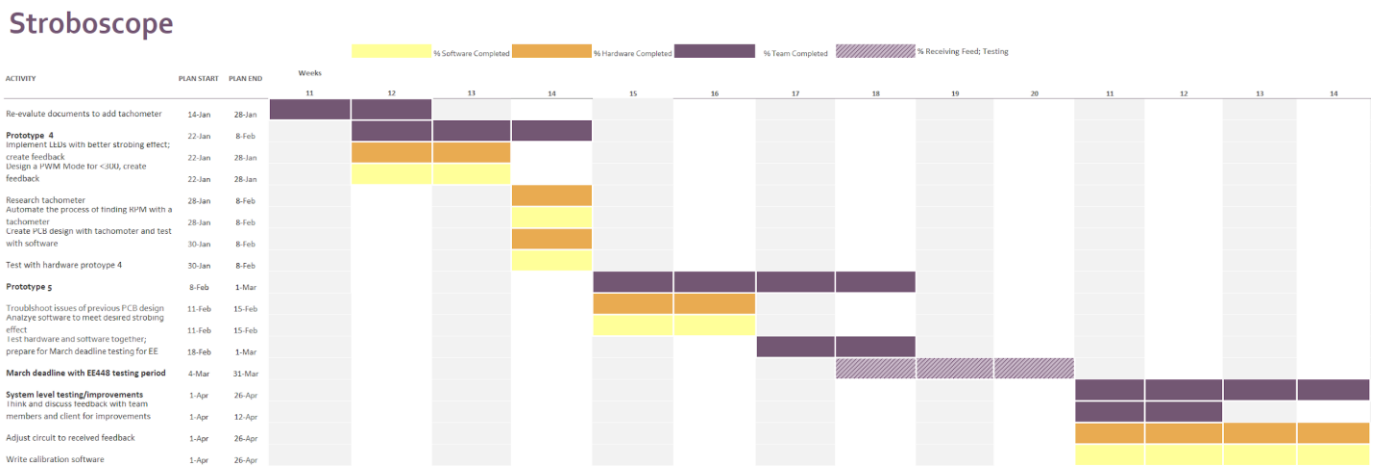
A. Circuit Schematic



B. Process Diagram



C. Gantt Chart (semester 1)



D. Gantt Chart (semester 2)