Stroboscope

DESIGN DOCUMENT

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

Revised: 12/3/18 - Version 2

Table of Contents

List of Figures

List of Tables

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 Introduction

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 and Project Statement

1.2.1 Problem Statement

EE 448 is a course on AC circuits and motors. In the labs of this course there are mounted motors that the students will do tests with and on, as seen in Figure 1.2.1.1.

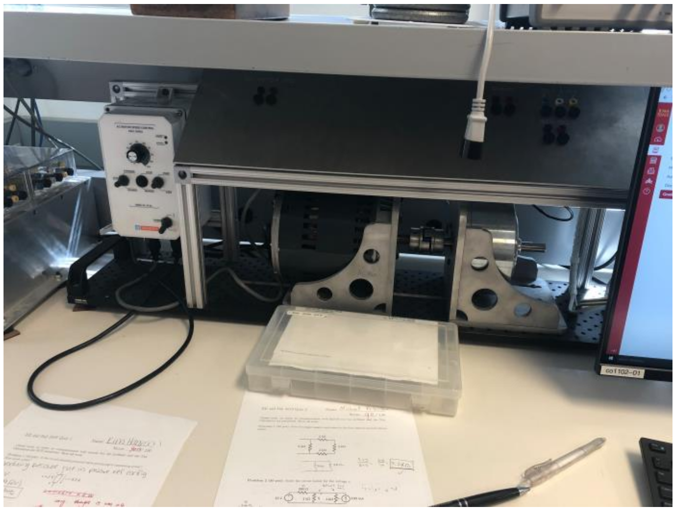


Figure 1.2.1.1

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 100-3000 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.

1.2.2 Solution Approach

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

The expected operating environment for the stroboscope is room 1102 in Coover Hall. It is a lab room for students taking the class EE 448. The stroboscope will operate under room temperatures, meaning there will not be many environmental obstacles. The system should be safe from 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 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 when they break. The ETG faculty have a strong background in engineering and fixing lab hardware, which means they will be competent users and will benefit from technical information which can help them debug and potentially improve the stroboscope. However, they are busy and will not be experts in this specific piece of equipment. Therefore, assembling the equipment should be straightforward, clear, and concise so they do not waste any more time than necessary on activities such as assembling the parts and debugging common issues.

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

1.5 Assumptions and Limitations

Assumptions:

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

Limitations:

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

1.6 Expected End Product and Deliverables

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

- Partially Functioning Prototype - December 2018
  - The first prototype will be used as a basic starting point to show to our client to determine if we have understood all of our requirements for the project properly. It will consist of a simple, functioning GUI that can be used for controlling the microcontroller to create the strobe effect with our LEDs.
- Fully Functioning Prototype - March 2019
  - The fully functioning prototype will have all the functionality that was in the partially functioning prototype as well as a more polished GUI with a modular LED and Tiva board. This prototype will be used by the students in the EE 448 lab, and we plan to use their feedback and testing 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's frame, 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.
        - Source Code - There will be frontend and backend code for both the GUI and the Tiva board being used. The code will be available on GitLab and provided to ETG staff.
        - 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 circuit and connections for the stroboscope.

2. Specifications and Analysis

2.1 Proposed Design

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.

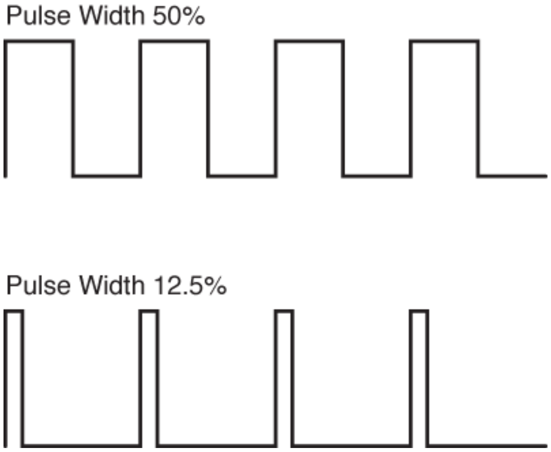


Figure 2.1.1 [2]

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.

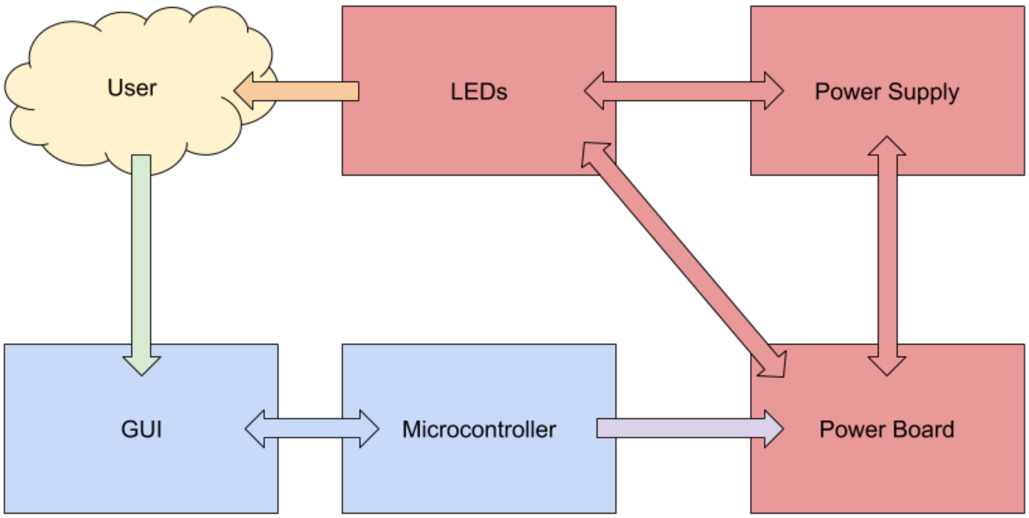


Figure 2.1.2

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 [3]. 24 bits allows for the microcontroller to count as high as 0d16777215.

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

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

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

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.

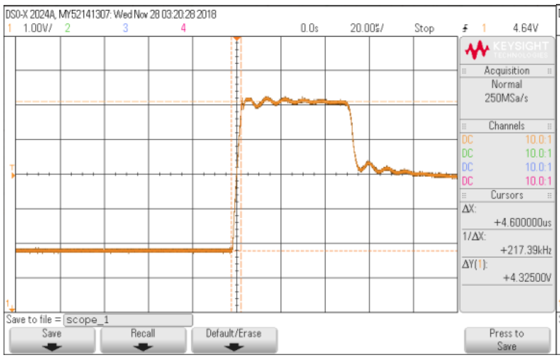


Figure 2.1.3

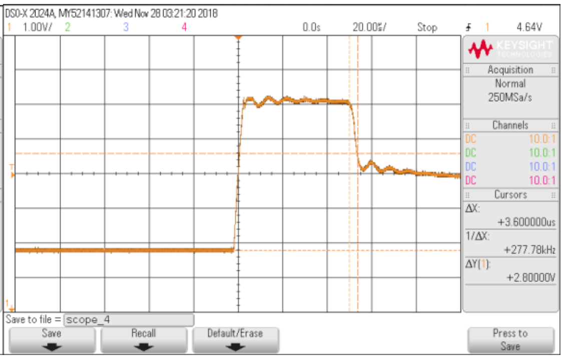


Figure 2.1.4

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).

Current Progress

At this point, we have implemented most of the basic functions (however, we have not yet calibrated them to be as accurate as we are expecting). Our GUI is implemented to take user input for the required functions. However, we have not yet implemented the pipe which allows it to communicate with the microcontroller. The microcontroller is able to output a modifiable waveform. At this point, we are hard-coding our period length value (however it will eventually be chosen based on the input from the GUI).

Our current circuit is able to take input waveform from a function generator to create a strobing effect. At the moment, it is easier to take input from a function generator because it is more configurable versus the software because as mentioned earlier, the microcontroller is still adjusted to output a modifiable waveform to the circuit. However, we can get a strobing effect with the software. We still have yet to implement test points to make sure none of the circuit components are overheating. Currently, our strobing effect isn’t as bright compared to the handheld stroboscope used the lab so we may research on different types of LEDs that are brighter for future prototyping.

Up to this point, we have a simplified version of a stroboscope working using a bench power supply and some extra parts from ETG to create a mock-up design that could prove our idea in practice while we waited on research and parts to come in from shipping.

With this setup we have an arduino with the default Blink library loaded into it which signaled a MOSFET that is being powered from the bench power supply that is lighting up our extra LED array that we salvaged from a magnetic work light. We have learned from testing these lights the bench power supply that we will need at most 2 amps at 5 volts from the power supply.

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

Requirements:

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

Non-Functional Requirements:

- 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

Standards:

Furthermore, 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 [4]
   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 [5]
   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 [6]
   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 [7]
   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.

2.2 Design Analysis

While creating our design, we have come to several key design options where we were able to identify multiple routes to take. To make informed decisions, we have created pro/con lists surrounding these design decisions. Seen below are the decisions which included the most factors and were therefore the most difficult to make.

First, in order to make a stroboscope, we have identified three possible ways to make the LEDs flash: a Tiva board, an Arduino board, and an AC circuit. All three options require a circuit to put together. After reviewing Table 2.2.1 with our advisor, we have decided to use a Tiva board. We have identified cost and function as the most important factors for our project.

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

Table 2.2.1: Pro/Con lists for design decisions

Next, we can choose between a modular design and a fully connected and soldered circuit. We created *Table 2.2.2* to help make this decision. We chose to design a modular board. Our advisor believes that the repair time would be a fair price to allow repair of the systems and therefore decrease the average cost of maintenance.

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

Table 2.2.2: Modular vs. singular board pro/con list

The stroboscope can include a physical knob to adjust the speed on the stroboscope, interaction taking place through a GUI, or both. We have decided that a knob would not make sense, because it would be the weakest part of the stroboscope in terms of durability and would provide minimal added convenience to the user. Therefore, our design will only include a GUI.

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

Table 2.2.3: Stroboscope control pro/con list

Finally, we need to choose software languages. We have already chosen the Tiva board, which uses C code. However, we need to decide what language to use for the GUI. Our advisor suggested using Python, so we have decided to evaluate whether Python will be suitable for our needs of if we will use a different language, such as Java. Based on the factors discussed in *Table 2.2.4*, we have decided Python will be sufficient because it is easy to learn, we would still need to learn how to complete all the functions in Java, and we have a resource for Python through our advisor.

Python		Java	
Pros	Cons	Pros	Cons
- Easy to use language - Good GUI class - Provided with example code from a different project	- No previous experience - Bad documentation	- Experience with Java - Experience using Java GUI classes	- IDE with which we have experience is for mobile applications - Will need to learn some functions still - No human resources

Table 2.2.4: GUI language pro/con list

3 Testing and Implementation

3.1 Interface Specifications

The first and main interface between the hardware and the software that will be used in testing is a square waveform that will be the output by the software on the Tiva board to control the LED circuit board. The software will output a square wave, which will be input into the hardware combined with our source of power to begin strobing our LED array. This will cause the LEDs to flash at the rate of the waveform, slightly impacted by the capacitors. We are using the capacitors to increase the brightness of the LEDs, which are dim from the short pulse width.

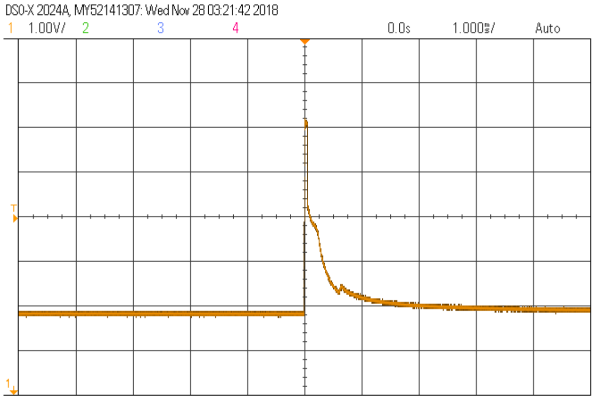


Figure 3.1.1 Current Signal at LEDs

Current testing shows that we are maintaining our pulse width within the turn-on voltage of our LEDs, which means that when the capacitor is draining we are not turning on the LEDs.

For later testing, we will add in test points where we can monitor the health of the circuit from the Tiva board and make adjustments to protect the equipment and the students. This will be in the form of pins, which the hardware sets the voltage to high to indicate a specific situation. The software will watch these pins and implement a form of a multiplexor to have an interruption that handles the problem being experienced by the hardware.

3.2 Hardware and Software

During the testing phase, hardware we have used has included the bench power supplies, function generators, and multimeters to accurately create a stroboscope to test the LEDs we have on hand and compare them to the current Stroboscope by brightness and switching speed. Since finding a good brightness, we have started adding LEDs until we have found a good amount of range in brightness at the same voltage level across all of the LEDs. This allows us to measure the amount of current we are drawing to find a suitable power supply.

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

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

The multimeter allows us to measure the turn-on voltage of the LED array and helps us determine the correct resistor value for our gate on the MOSFET. Without the correct resistance we will not have high enough voltage for the LEDs to turn on from the power supply. On the other hand, too much voltage can cause permanent damage on the LEDs.

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

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

3.3 Functional Testing

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 [8, 9], 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.

3.4 Non-Functional Testing

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.

3.5 Process

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

For the development and test of our stroboscope, we have developed the process which can be found in *Appendix B.* For each prototype, we will start by identifying the problems we are trying to solve with this prototype. Then, we will brainstorm ideas to solve these problems. After this, we will develop the new prototype hardware and software components. Finally, we will test the prototype. We will start with hardware and software testing, and once that is complete move on to system level testing. At this point, if our final evaluation test criteria have been satisfied, we will be done. If not, we will create a new prototype.

3.6 Results

When testing the transistor for functional requirements we were pretty successful. We conclude from the testing that the transistor we chose is satisfactory for our needs. We simulate our circuit using a bench waveform generator.

Modeling and Simulation

When we were modeling our power and LED circuit in Ultiboard, we used the integrated PSpice functionality to view our waveform before fully testing new hardware. This shows us the estimated wave before breadboarding the new hardware.

We found some limitations with the simulation in PSpice because of the feature-less oscilloscope that didn't allow us to modify the waveform to our actual testing specifications. Therefore, we set the pulse width to 1% and measured off of the first pulse from the LEDs.

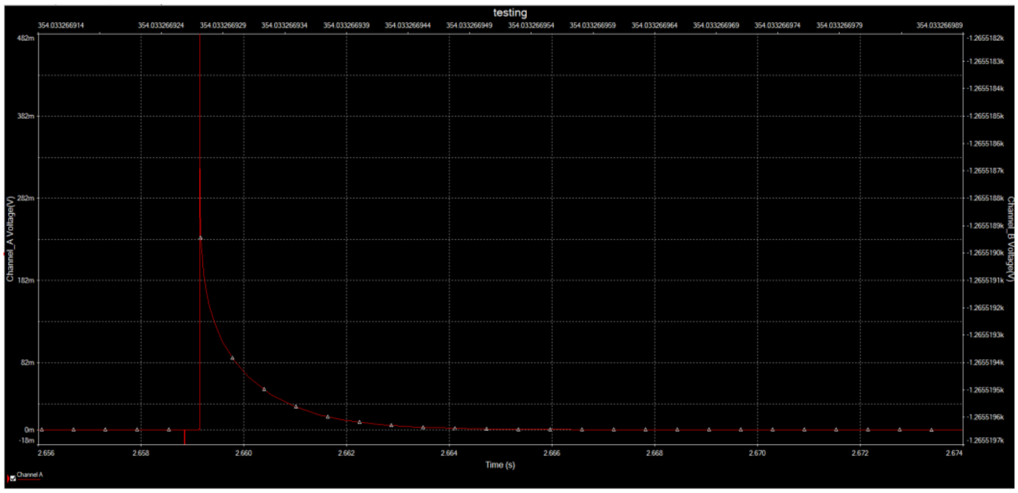


Figure 3.6.1 Circuit Simulation
This was mimicking the circuit waveform from 3.1.1

When modeling our circuit for the first printed version, we used Ultiboard again. It allowed us to visualize the components we are going to be using and showed us the spacing and relative size of each component shown below.

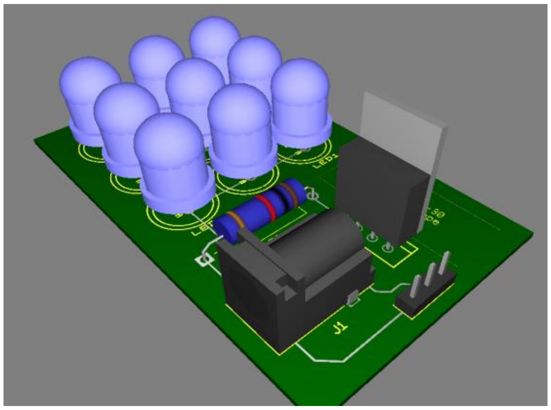


Figure 3.6.2 Circuit Modeling

Implementation Issues and Challenges

On the software side, we have run into some issues and challenges. One of the issues we have right now is figuring out a good way to interface between the GUI and the Tiva Board. We have done some research and tested some different methods to see if any kind of connection works, but we have not got anything to work well for that interface. We plan to do more research into this and figure out the best way to get this issue figured out, however this process has been slow because Python is not well documented online. We are looking to identify an in-person resource. Another issue we will be running into on the software side is getting the Tiva Board to have a speed of 100 RPM. Our issue here is that we need to have a long period to get this to work, but a long period is more difficult with a fast processor. Due to the processor speed of 80MHz and the number of bits reserved for PWM mode, we can only go as low as 300 RPM with the built-in PWM mode. We will be writing our own PWM mode to go from 100 to 300 RPM.

On the hardware side, we have run into some issues and challenges. One of the issues we have run into is making the LEDs bright enough. In order to make them brighter, we have to change the circuit, but by changing the circuit, we will be affecting the signal; therefore, it is challenging to make changes to the circuit in order to increase the brightness without affecting the signal too much. Another issue we have run into on the hardware side is getting the strobe effect to work. Our issue was that the pulse width for the PWM mode was not long enough to get the strobing effect. This will still be an issue as we keep trying to get the brightness to be correct because as the brightness increases, so does the pulse width. Therefore, if the pulse width gets too wide, the stroboscope will no longer strobe. Our challenge will be to get the right brightness for the stroboscope without causing the pulse width to get too wide.

4 Closing Material

4.1 CONCLUSION

Our main goal is to create a stroboscope that is mounted, modular, and user friendly to withstand over time, have more cost effective repairs, and provide easier functionality for students in the lab. Through various testing and prototyping, we hope to achieve a stroboscope that can satisfy all the needs of the students, our client, and ETG.

We hope our design is successful in fulfilling all the criteria our client, Matt Post, has requested. The students of EE 442 and EE 448 should find our stroboscope more user friendly and less breakable making the lab itself be less frustrating and less of a hassle to complete. If ETG is to make any repairs, they should be easier than the previous model because of the modular design and more cost effective since the parts are in house.

The team's main focus is to make sure these requirements are met and provide a safer, more cost effective, and more efficient device.

4.2 References

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

[2] "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].

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

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

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

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

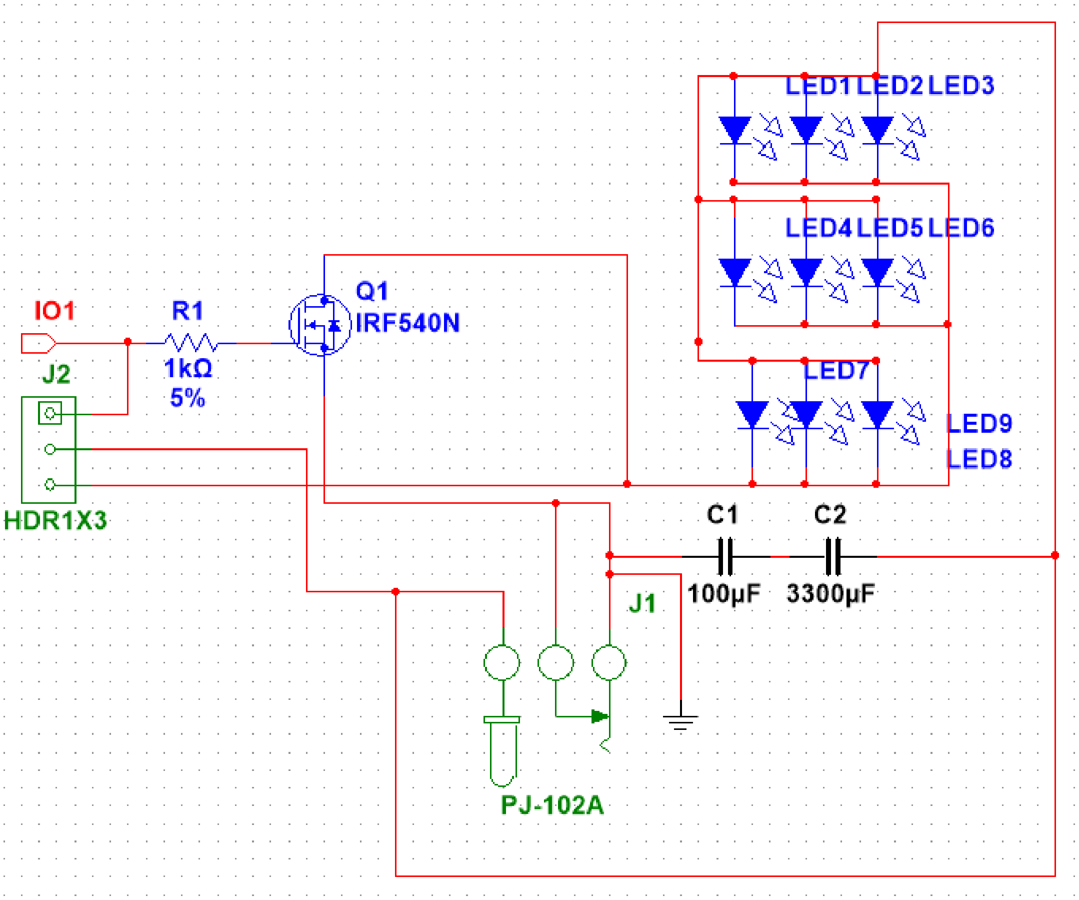
[7] *IEEE Recommended Practices for Modulating Current in High-Brightness LEDs for Mitigating Health Risks to Viewers*, IEEE Standard 1789, 2015.

[8] 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.

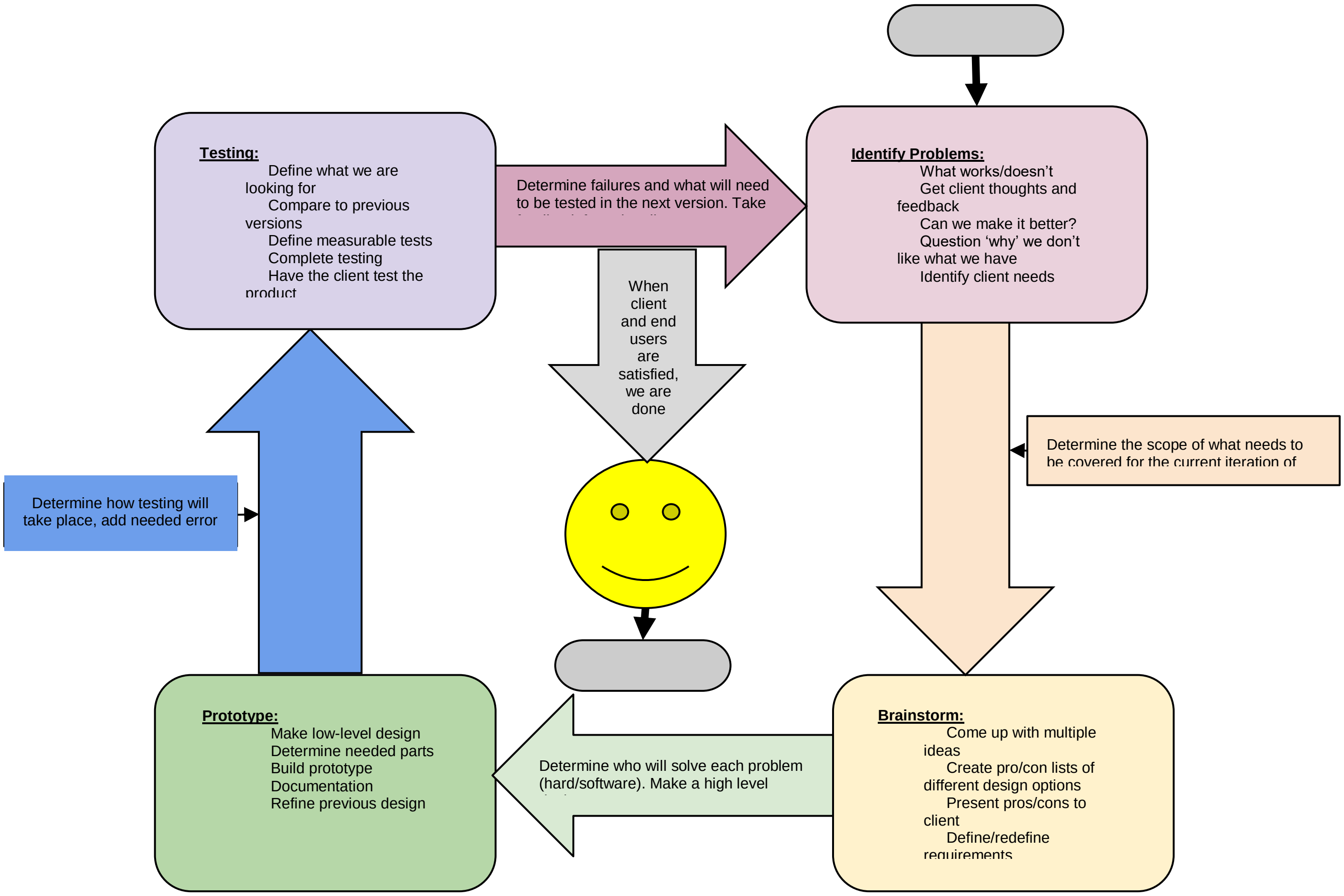
[9] T. Bigelow. "EE 448 Lab 6 Report.doc." Unpublished manuscript, EE 448: Introduction to AC Circuits and Motors, Iowa State University, Ames, Iowa, United States.

4.3 Appendices

A. Circuit Schematic



B. Process Diagram


Testing:
Define what we are looking for
Compare to previous versions
Define measurable tests
Complete testing
Have the client test the product
Determine failures and what will need to be tested in the next version. Take
Identify Problems:
What works/doesn't
Get client thoughts and feedback
Can we make it better?
Question 'why' we don't like what we have
Identify client needs
When client and end users are satisfied, we are done
Determine the scope of what needs to be covered for the current iteration of
Determine how testing will take place, add needed error
Prototype:
Make low-level design
Determine needed parts
Build prototype
Documentation
Refine previous design
Determine who will solve each problem (hard/software). Make a high level
Brainstorm:
Come up with multiple ideas
Create pro/con lists of different design options
Present pros/cons to client
Define/redefine requirements