

Portable BCI Stimulator

ECE 445 Spring 2013
Design Review

Team 17: Bonnie Chen, Randy Lefkowitz, Siyuan Wu
TA: Ryan May
February 12, 2013

Contents

1. Introduction	3
2. Design	4
2.1. Overall System	4
2.2. Block Diagram	5
2.3. Block Descriptions	5
2.3.1. PC/Wireless Transmitter	5
2.3.2. Wireless Receiver	6
2.3.3. Microcontroller	6
2.3.4. LED Array	6
2.3.5. Power	6
2.3.6. 2.3.6 User	6
2.4. Schematics	7
2.4.1. TLC-5940 LED Driver	11
2.4.2. Wireless Bluetooth SMD Module	12
2.4.3. Arduino Uno R3	12
2.5. Simulation and Calculations	12
2.5.1. TLC-5940 LED Driver Calculations for Power Dissipation	12
2.5.2. LED Luminosity Calculations	13
2.5.3. Power Budget	15
3. Requirements and Verifications.....	15
3.1. Testing Requirements and Verifications	15
3.1.1. PC and Bluetooth Transmitter	15
3.1.2. Bluetooth Receiver	16
3.1.3. Microcontroller	17
3.1.4. LED Array	18
3.1.5. Power Supply	19
3.2. Tolerance Analysis	20
3.3. Ethical Issues	21
3.4. Safety Considerations	21
3.5. Contingency Plan	22
4. Cost and Schedule	22
4.1. Cost Analysis	22
4.2. Schedule	23
5. References	26

1. Introduction

Brain Computer Interfaces (BCI) based on Electroencephalography (EEG) allow for the monitoring and analysis of ongoing brain activity in real time. The signals measured by this technology can be used to control user interfaces without the requirement of the human motor system. This technology can benefit those with paralysis and other severe disabilities. As of now, the majority of BCI systems are currently large and immobile, and therefore impractical for use in everyday life outside of a lab. There are several components to a BCI system such as data acquisition, a classification system, as well as stimulation, all of which must be made portable to create a portable BCI. To address this problem, we would like to focus on making a portable stimulator that can interact, through wifi, with the BCIs that are monitoring brain activity. The stimulator will consist of flickering LEDs at predefined frequencies, with attention to luminescence (we don't want our LEDs to blind the user so it must be at the right intensity for each user) as well as controls to adjust the frequencies while maintaining signals timing. Our design goals are to make the stimulation for the BCI and EEG portable and be integrated wirelessly so that users are not confined to just a lab setting and that the system could be tested and used in different environments.

Purpose

Most brain computer interfaces (BCI) are limited to laboratory settings. We would like to make a BCI stimulator based on electroencephalography (EEG) that is portable, and therefore useful in real world applications. This could be a great advancement in allowing people with paralysis to communicate without any type of movement.

Features and Benefits

The circuit design will be made compatible for a glasses-mounted stimulator, which will include proper mounting for 10 LEDs, 5 for each eye, creating convenience to the user and adding to the portability of the EEG and BCI system.

The stimulator receives the LED blinking frequencies wirelessly through Bluetooth and user inputs into a PC program. Wireless communication with the PC will enable the stimulator to be operable from a distance of around 30 ft.

Flashing lights can induce pain to the user's eyes. The stimulator will prevent this by giving the user controls to vary the frequency and duty cycle of each LED. The frequencies of each LED controlled by the user can be variable from 1 to 100 Hz, which are within the useable operating limits of EEG detection. The duty cycle of each LED can be variable from 10 to 90%, which gives the user sufficient control of light intensity coming from each LED.

2. Design

2.1 Overall System

The way the stimulator is implemented in the overall BCI and EEG system is shown in Figure 1. The brain signals from the user are being monitored by the EEG and is amplified and sent to the computer for Data Acquisition and Signal Processing. After this the signal is sent to the User Interface, which takes inputs for the controls to the stimulator. The control inputs from the user are transmitted to the stimulator from the computer and the resulting changes are fed back to the user being monitored with the EEG.

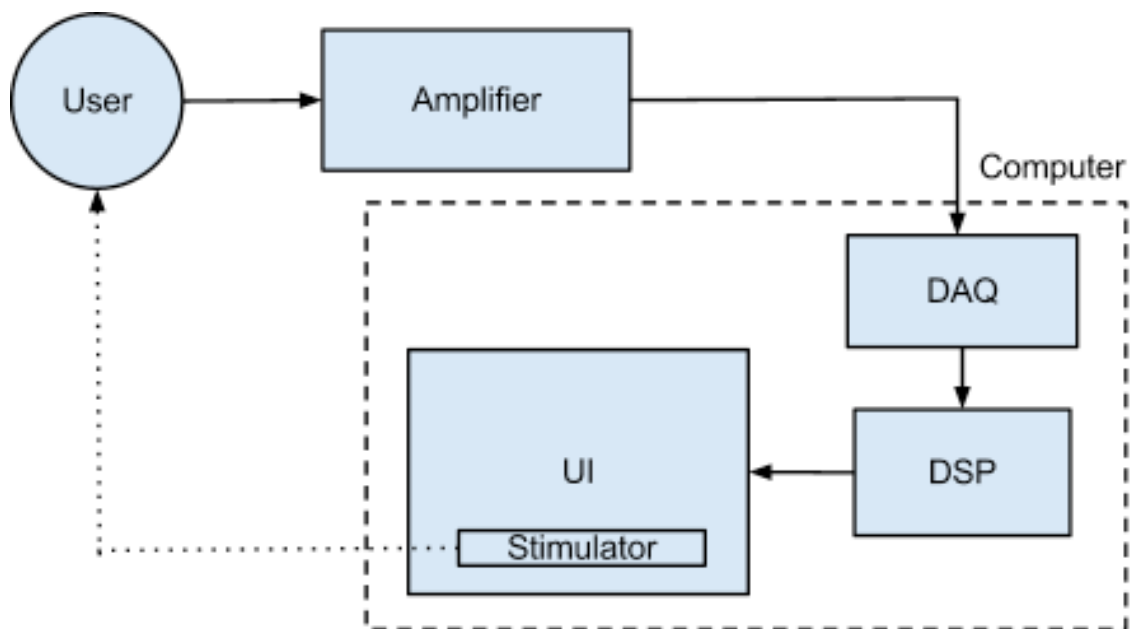


Figure 1. Overall BCI Schematic

2.2 Block Diagram

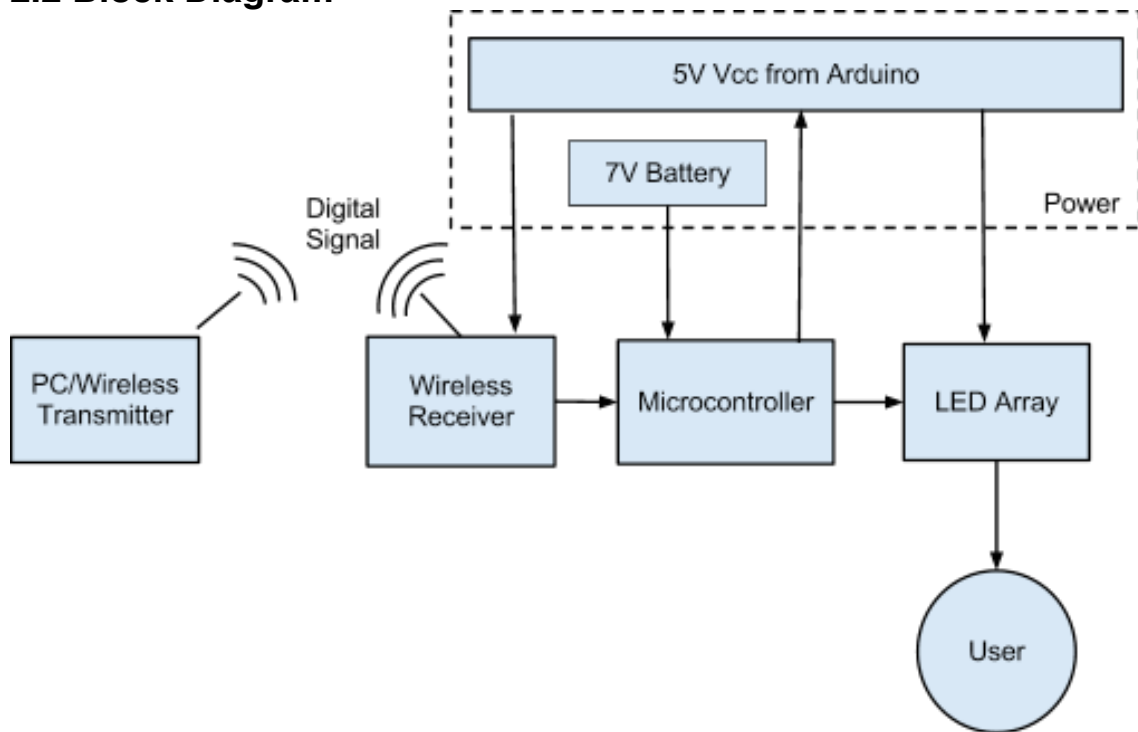


Figure 2. Top Level Block Diagram for Stimulator

2.3 Block Descriptions

2.3.1 PC/Wireless Transmitter:

The PC will load the program, which can control the frequency and duty cycle of the LED array. This will include input from the user to change the frequency of each LED on PC and have the resulting information be transmitted to the microcontroller through the wireless components. The wireless transmitter is contained in the PC.

The wireless transmitter will be used to send serial data to the microcontroller that controls the LED array. This will be the bluetooth transmitter within the computer using bluetooth v2.0 protocol.

2.3.2 Wireless Receiver:

The wireless receiver is a UART serial TTL bluetooth module. It has 4 channels: an input of 5V, Ground, TX, RX. Both TX and RX have 3.3V outputs.

Communication is set up by connecting the RX channel of the receiver to the TX channel of the arduino and the TX of receiver to the RX of the arduino. After the pairing is done through programming on the PC, data is sent through command lines on the pc to the Arduino. The wireless receiver will receive and interpret the multiple packets of serial data sent from the PC as if the two were connected by

a wire, and translate them into a signal for the microcontroller to work with in outputting the correct frequency.

2.3.3 Microcontroller:

The microcontroller used for the stimulator will be the Arduino UNO R3. The controller input will be serial data from the wireless receiver and will output the frequencies and luminosities chosen and set by the user through the program to the LED driver through the PWM and Digital Output pins. It will be powered by a 7V source.

2.3.4 LED Array:

The LED array will consist of an LED driver (TI TLC-5940) which will send the proper PWM signal outputs to 10 LEDs, 5 on each eye. The TLC-5940 is powered by a maximum of 5V and outputs to each channel a voltage of 2-3.2V which is the necessary voltage to light up the LEDs. The LED driver receives serial data from the microcontroller module and feeds the data to a connected array of 10 LEDs no bigger than 5mm, which will later allow for easy mounting onto a wearable frame and provide for adjustments in luminosity and light color. The LEDs have a 2-3.2 voltage drop across each and a maximum forward current of 20mA, in which if necessary, limiting resistors will be used to protect the LED's from burning out. But because the LED driver handles current and voltage output for each LED, the LEDs will be given power following the proper power dissipation procedures outlined by the TLC-5940 data sheet (see calculations for power dissipation of LED driver in 2.4.1). The LEDs will output the final blinking frequency to the user.

2.3.5 Power:

All portable components in the stimulator will require power from this module. The microcontroller will be powered by a 7.4V lithium ion battery. Because of the voltage regulators in the Arduino Uno, the other components will be connected through the 5V output pin on the Arduino. The TI TLC-5940 LED driver and the Bluetooth module have maximum input voltages of 5V, which is what we will be used to drive 10 LEDs.

2.3.6 User:

When the user is looking at a certain LED with a certain blinking frequency, information about the output brainwave signal from the user will be retrieved by the EEG/BCI system and will determine which LED the user is looking at. Because this is handled solely by the EEG and BCI, we will not be testing this portion while testing the stimulator.

2.4 Schematics

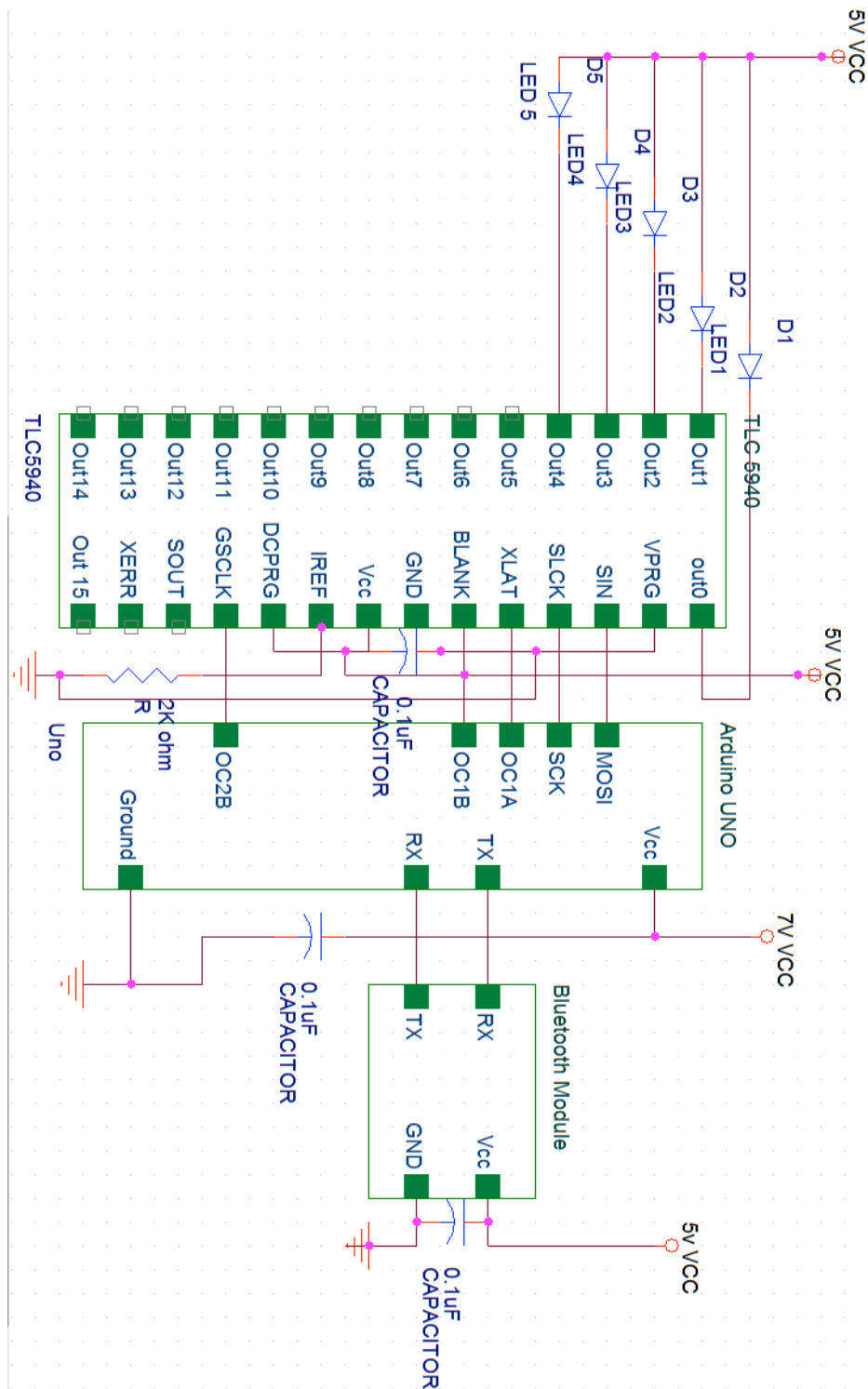


Figure 3. Schematic for the Stimulator Circuit

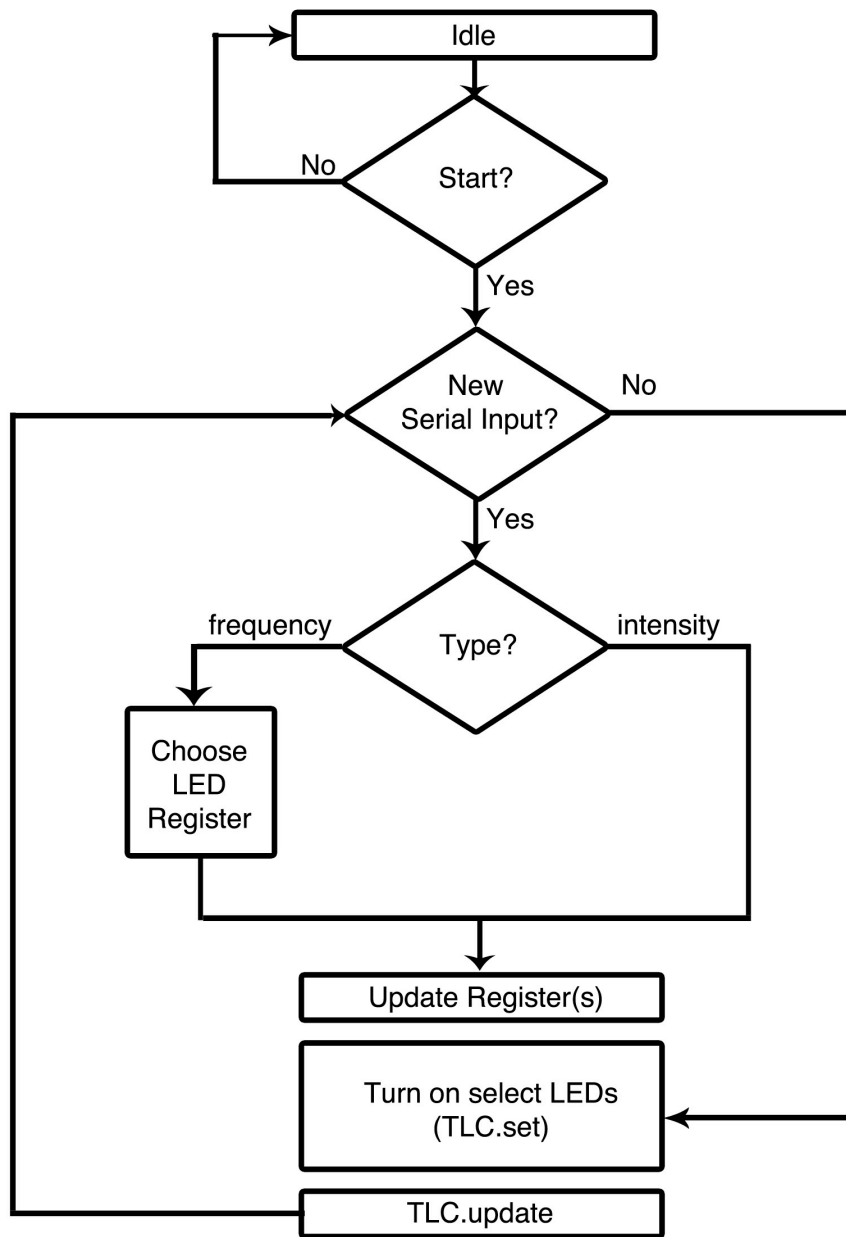


Figure 4. Arduino Code for Serial Flowchart

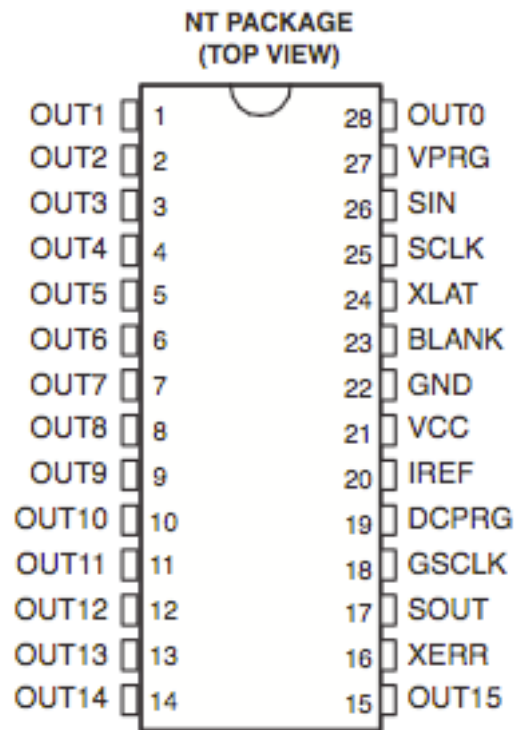


Figure 6. Texas Instruments TLC-5940 NT [1]

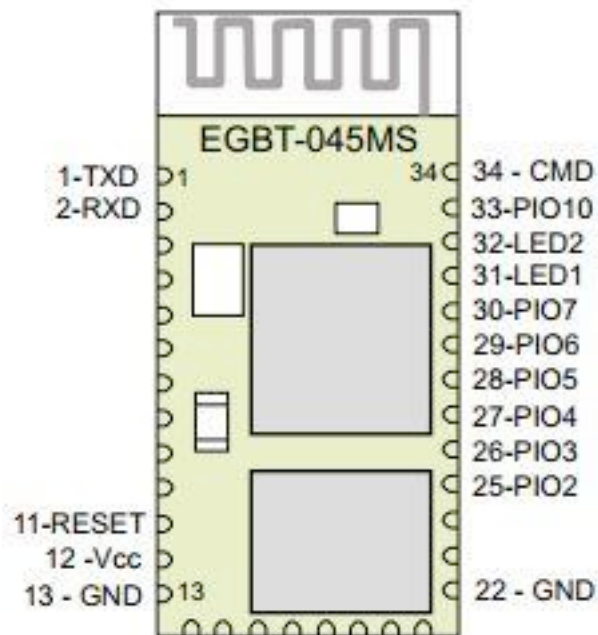


Figure 7. EGBT-046S Bluetooth Module [3]

2.4.1 TLC-5940 LED Driver

The TLC-5940 LED driver (Figure. 6) requires 5V external voltage applied on pin Vcc to power it. It has 15 output pins to drive the LED with a constant current source. The magnitude of the current source will be set by the current on pin IREF, which will be connected to a 2k ohm resistor and gives us a current of about 20mA to drive the LED.

Below is the configuration of pins connected on TLC-5940 based on the tlc5940arduino google code library [7]:

IREF: The current through this pin will determine the magnitude of current source

in all the output pin. From the formula
$$I_{MAX} = \frac{39.06}{R_{iref}}$$
 in order to drive the LED at 20mA current, we need to put a 2K ohm resistor in this pin.

out0-out4: Output pin for the LEDs. All the LED will be connected to 5V Vcc and the output pin. Current source depends on Iref.

SIN: Serial input of TLC-5940, connected to the MOSI (Master Out Slave In) pin on the arduino board. This is where the data from the arduino (when LEDs should be on/off, as well as intensity) will enter the TLC-5940.

SCLK: This is the clock for the shift register that shifts in new data. It will be connected to the SCK (serial clock) pin of the Arduino which is the clock to synchronize data transmission generated by the master.

XLAT: When XLAT = high, the TLC-5940 writes data from the input shift register to either GS register (VPRG = low) or DC register (VPRG = high). In our case, since VPRG will be connected to ground, data will be written to GS register.

When XLAT = low, the data in GS or DC register is held constant. XLAT will be connected to OC1A, since there is a library of code for TLC-5940 that use OC1A to control the XLAT signal.

GSCLK: This is the clock for the GS register, which determines the speed of the TLC's PWM output, thus controlling the LED brightness. It is connected to OC2B on the Arduino. This is a timer that we set to control the PWM.

DCPRG: Selects the source of the current limiter register, it will be high all the time.

BLANK: Marks the end of a PWM cycle in addition to blanking the output. We will reprogram TIMER1 to generate this signal. That will cost us the native PWMs on digital 9 and digital 10.

2.4.2 Wireless Bluetooth SMD Module

Vcc will need an external 5v power supply for the bluetooth module (Figure 7)
TX/RX: Used to receive (RX) and transmit (TX) TTL serial data. These pins will be connected to the corresponding TX/RX pin on the Arduino board.

2.4.3 Arduino Uno R3

All the pins that will be connected have been explained above by the TLC-5940 and the bluetooth module connections. Vcc will be powered by a 7.4V Lithium Ion Battery (Figure 5.)

2.5 Simulation and Calculations

2.5.1 TLC-5940 LED Driver Calculations for Power Dissipation

The TLC-5940 has to have the proper power dissipation in order to sustain and prevent the chip from overheating. Because it is in charge of driving 10 LEDs, the power dissipation of the LED driver will be of high importance if we are to ensure that the circuit will not fail and that the driver will not be overheating while being worn by the user. Using the given equation from the TLC-5940 data sheet [1] (eqn. 7) we can find the power dissipation given off by the circuit

$$P_D = (V_{cc} \times I_{cc}) + (V_{OUT} \times I_{MAX} \times \frac{DC_n}{63} \times d_{PWM} \times N) \quad \text{Eqn. 1}$$

$$V_{cc} = +5V \text{ for Arduino}$$

$$I_{cc} < 25mA \text{ for Data transfer of 30MHz}$$

$$V_{OUT} = 5V - 2V = 3V \text{ where 5V is the voltage going into the LED driver and 2V is the voltage drop across a red LED (3.2V for green and blue)}$$

$$I_{MAX} = \frac{39.06}{R_{iref}} \rightarrow \text{Using } 2k\Omega \text{ for a maximum output current of 20mA to each LED channel}$$

$$DC_n = 63 \text{ by default}$$

$$d_{PWM} = \frac{PWM_{set}}{PWM_{max}} \rightarrow \text{which is 4096 from LED Driver settings}$$

$$N = 10 \text{ for each LED channel we will be assigning a PWM value to}$$

For the TLC-5940 we will ensure the LED driver will operate safely at a power dissipation value less than the maximum power dissipation specification for the 28-pin HTSSOP with PowerPAD, which is 2026mW.

Our duty cycle will never be above 50% so the d_{PWM} will never exceed a value of 0.5. Thus our maximum power dissipation for the TLC-5940 being used as the LED driver will never be greater than 575mW, which is less than 28.3% of

the maximum power dissipation rating of the driver. We will ensure this during testing that all the voltage and current values for the above equation fall within a 5% accuracy to the values we used to calculate the maximum power dissipation.

2.5.2 LED Luminosity Calculations

The LEDs' luminosities will be tested and recorded for determining operating intensities that provide the most user comfort. This means a proper conversion of measured power from the LEDs intensity on a photodiode to a value of Luminance in which we can use to determine how much strain the eye is receiving from the LEDs' brightness.

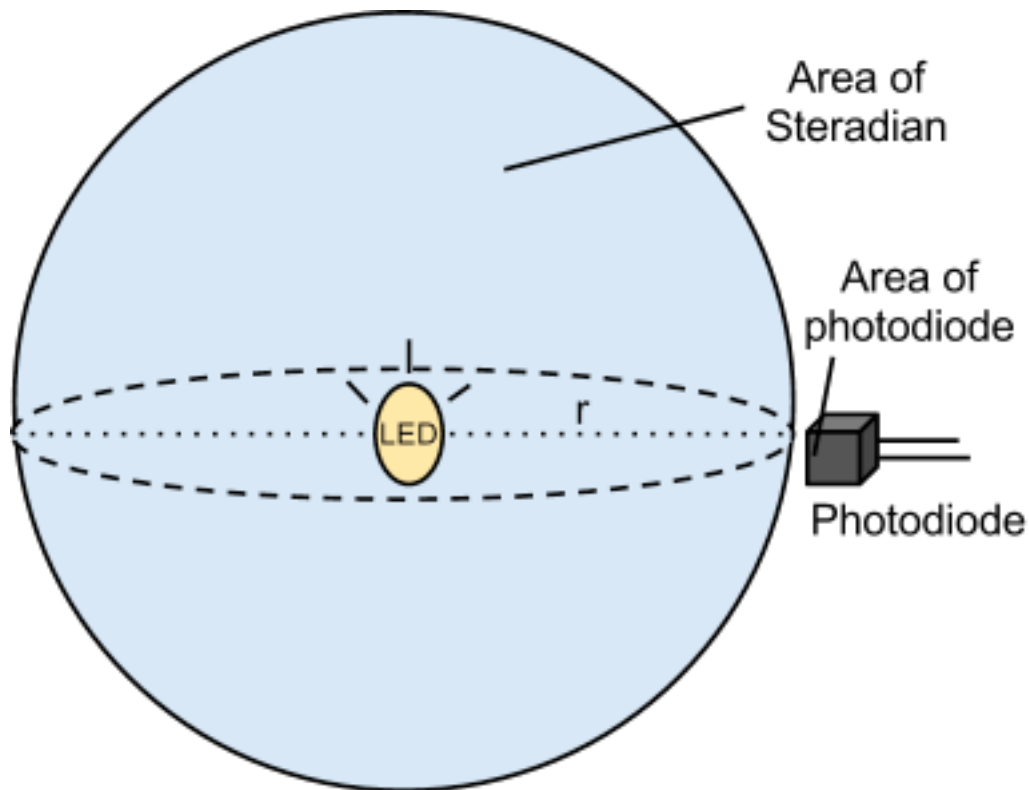


Figure 8. LED Intensity Capturing

The testing process will first require us to use a photodiode with an area of around 4mm (roughly the average size of the pupil) to measure the output current with which we use to find the power incident on the photodiode from the following equation.

$$\frac{I_{measured}}{S} = P_{measured} \quad \text{Eqn. 2}$$

where S is the measured sensitivity of the photodiode (for silicon photodiodes, this value is usually around 0.43A/W [5]). The photodiode will be placed at a distance r from the LED as shown in Figure 8. This distance should correspond to how far we will be placing the LEDs from the eye when mounting it onto the stimulator (which will be ~50mm).

We can measure the Power needed to provide 1 cd on the LED, P_{active} , by finding

$$\frac{A_{photodiode}}{A_{steradian}} \times P_{cd} = P_{active} \quad \text{Eqn. 3}$$

where:

$$A_{steradian} \approx r^2$$

$A_{photodiode}$ is the size of the photodiode's receiving area [$\sim mm^2$]

And P_{cd} is the power of the LED equal to 1 cd, measured in Watts per Steradian which can be found by the following equations

$$ScotopicConversion \times \frac{\lambda_i}{\lambda} = P_{cd} \quad \text{Eqn. 4}$$

where:

Scotopic Conversion is a value in units of Lumens per Watt and can be found by the use of a Luminous Efficacy Table [4]

λ_i is the wavelength of the LED provided by the LED datasheet

λ is the wavelength given from the Luminous Efficacy Table (varies based on the color of the LED being used)

The final step is to divide the power measured by the power to provide 1 cd on the LED, combining equations 2 and 3

$$\frac{P_{measured}}{P_{active}} = N \quad \text{Eqn. 6}$$

where N represents the candela measured on the photodiode's active area, which is an approximation of how much candela the user's pupil is receiving from the LED.

The number of candela can tell us at what luminance the LED's should be at before the user's eyes experience discomfort or severe eye strain occurs. For blinking LEDs in relatively low light settings (inside a building), the luminance contrast between the LED intensity and the surrounding background should be no more than 20 cd apart [6].

2.5.3 Power Budget

As a portable device, the user will need to be capable of using the device for long periods of time without the need to reconnect the stimulator into a power source. This means that not only should the stimulator be a well designed and compact device, it should be efficient and sustainable in power. Using a 2200mAh rated Lithium Ion battery, we can approximate the usage time for the stimulator using the maximum possible current draw and voltage requirements for each modular component in the system.

Component	IMAX (mA)	Voltage Requirement
Microcontroller	40* (2 output pins)	7-12 V (ideal)
LEDs	20 * (10 LEDS)	2V (red), 3.2V (green, blue)
Bluetooth Module	40	5V
LED driver	120	Vcc > 3.3V
Total	440	-----

Usage time = [2200 mAh] / [440 mA] \approx 5 hours of charge

3. Requirements and Verifications

3.1 Testing Requirements and Verification

3.1.1 PC and Bluetooth Transmitter

Requirements	Verification
1.) Wireless communications must be configured through the PC to ensure proper transmission of the data	1.) For pairing the device, we find the bluetooth device in control panel and add the bluetooth device to PC. If it shows successfully added, it is connected to the PC.
2.) PC must be able to take in channels (LEDs or intensity) and values	2.) We will test that the input is being taken from the user, validated, and

(frequencies or intensity values) from users, verify that they are valid, and store them.	stored by inputting values we know are invalid and observing the expected error prompt, as well as giving valid values that we will then print back to the screen.
3.) The computer must translate the stored values into serial data.	3.) When a value is changed, the code will change the value of our 11 bit output variable. It will make 1 start bit, convert the channel into an address (next 3 bits), and the value of that channel into the last 7 bits. We will then print the new computer output to the screen and verify it is correct.

3.1.2 BlueTooth Receiver

Requirements	Verification
1.) Transmitter from the PC must be supported by the bluetooth receiver	1.) Verify the computer has a Bluetooth transmitter that is either Bluetooth version 2.1+ or 2.0, 1.2, 1.1. Data will be sent to the bluetooth receiver from the PC through PUTTY. Choose serial as connection type. Pick the right COM connection port and make the connection.
2.) Testing if the bluetooth receiver has received the data sent from the putty	2.) To test this, we will run program on the Arduino that turns an LED on when we send a 1 from the PC, and off when we send a 0.
3.) TTL Bluetooth Receiver must be able to receive data transmitted at a frequency that will not interfere with the operating frequencies for the LEDs	3.) After the receiver-end code on the Arduino is done, we will test and see if sending frequency signals that are higher than the upper bound operating frequency (100Hz +) to ensure that the transmission frequency of bluetooth of ~2.4GHz will not affect the overall signal in functioning properly,

	monitoring the received data with an oscilloscope.
--	--

3.1.3 Microcontroller

Requirements	Verification
1.) After the testing for bluetooth module is done, we will now test if the microcontroller has receive the correct data from bluetooth module.	1.) To test this, since there is not much display to show what was received, we will test this part by viewing the data on the receiving pin using the Arduino IDE. For example, now we want to set LED0 to frequency 2HZ. The decoded data seen on the pin should be 1 000 0000010 start LED Value
2.) We will test that the serial data is properly decoded, and the correct values are stored in our variables.	2.) We will input a known 11 bit value to the decode function, and we will print out the values of our 5 LED and 1 intensity registers to verify that the correct value has been stored in the correct location.
3.) A counter will operate at Arduino's clock frequency. The code will keep track of which LED should be on or off at each clock pulse.	3.) We will verify the counters speed by looking at its value in the IDE. We will also look at each LED's value over 1 second on the IDE, and make sure that they are on and off the correct number of times based on their frequencies. (Ex: LED0 has a value of 5 Hz and intensity=50%. LED0 should switch between 0 and 2048 5 times over 1 second).
4.) The data then needs to be output in a way that the TLC5940 can interpret (serial) using the Arduino's tlc library (tlc.set and tlc.update) to send the data.	4.) We will verify that each time an LED value changes, tlc.set is called, and tlc.update is called at the correct intervals.

3.1.4 LED Array

Requirement	Verification
1.) The serial data sent to the LED driver will output the correct PWM signals to the desired output pins (10 total)	1.) We will set our GSCLK to a predetermined frequency, then change the value of GS register 1, and by monitoring the output of channel 1 on an oscilloscope, we will verify that the duty cycle being seen is the correct fraction of the 4096 count grayscale cycle (2048=50% of GSCLK, etc).
2.) LED driver must provide proper power dissipation to properly handle and control every LED	2.) Monitor the current output by the TLC-5940 as well as the voltage across the LEDs. Make sure that the current does not exceed 25mA which is the rating we are basing the R_{iref} value of $2k\Omega$ off of and that the voltage value is no greater than 3.2 V which is the maximum voltage to power the LEDs and no less than 2V which is the minimum voltage to power the LED
3.) Each LED must operate with an intensity that will not cause discomfort to the user	3.) Verification for the LED intensities will consist of finding safe operating limits and environments in which the stimulator can be used. This will be done first qualitatively by us looking at the LEDs, but to insure that this is achieved, there must be a proper recording method of finding the luminance values for the LEDs which will require a photodiode to measure different intensities. <ul style="list-style-type: none"> a. The LED intensity will be measured with a photodiode and amplifier (optional). The photodiode (~4mm in diameter, the average size of the pupil) generates a current found in eqn 2. to give us the Power present on the photodiode at distance ~50mm from the photodiode.

	<p>This is equivalent to the distance we are mounting the LEDs away from the user's eyes. Using equation eqn 6. we can find the number of cds the LED is giving off at that distance.</p> <p>b. If the current signal generated by the photodiode is too small, add an amplifier circuit to measure the current.</p> <p>c. Test and measure several light intensities by changing the PWM value for each LED channel, record and measure which light colors and levels of intensity are the most comfortable to the user (see section 2.5.2)</p>
--	--

3.1.5 Power Supply

Requirement	Verification
<p>Sufficient power must be provided for every portable component</p> <p>1.) Bluetooth</p>	<p>1.) The power for the Bluetooth will be coming from the Arduino 5V output pin. We will need to test for a stable open circuit voltage output from the Arduino 5V pin using a multimeter connected from the output pin to ground and make sure the voltage does not vary over 5% to ensure proper power to the bluetooth module, anymore than that can affect the behavior of the wireless transmission.</p>
<p>2.) Microcontroller</p>	<p>2.) The Arduino Uno will be powered by 7.4V battery, with which we will measure using a multimeter connected</p>

	<p>from the positive battery terminal to ground to ensure that no less than 5% of that value is coming out.</p>
<p>3.) LED Driver</p>	<p>3.) The LED driver will have an input voltage of 5V from the Arduino, with which we will measure with the multimeter in the same fashion as to how we measured the power for the Bluetooth module.</p>
<p>4.) LEDs</p>	<p>4.) We will test the currents drawn by the LEDs with a multimeter and make sure that no less than 0.5mA of the required forward current is being measured. Any less will probably not be enough to power the LED and will be noticed if it is not turning on, in which we will need to step backwards in the power testing and look at the driver and see if it is outputting the correct constant output current which is 120ma.</p>
<p>5.) Power supply must be efficient in delivering power that will allow maximum usage time of the stimulator for the user.</p>	<p>5.) Find the current draw at multiple loads using a multimeter, which should fall within 5% of the specified amps given by the data sheets, compare power usage with the given values for the current and voltages to the rated mAh's on the batteries to determine usage time for the stimulator (see section 3.5.3)</p>

3.2. Tolerance Analysis

For this project, the user end on the PC must be working in order to send data to the Arduino. Also, the frequency of each LED must be controlled individually by the user. The LED array must be precise with the duty cycle and frequency of the LEDs, because all the data gathered by the BCI and EEG system is in real time.

The proper tolerance range of the frequency output is the desired frequency $\pm 0.5\text{HZ}$. The LED array must be able to be mounted on the glasses mount.

One way the duty cycle and frequency can be affected by other parts of the circuit is by the wireless transmission of our controls from the computer. Thus, extensive testing and analysis of the tolerance of the frequency and duty cycle that is affected by the information transmitted by bluetooth in real time must be accounted for in our design of the stimulator.

Also because the final output of our stimulator system is into the user's eyes, we must preserve the comfort and visual perception of the user, meaning that the LED array must account for not only precise frequencies and duty cycle output, but also safe operating limits for various users such that no discomfort or harm will be done onto the user's vision through the use of the blinking lights. This will require proper testing and operating limits that are safe for long periods of exposure to blinking lights to the human eye.

3.3. Ethical Issues

This project has an extraordinary range of possible applications in the future. With many applications come many possible ethical issues. For starters, blinking lights in people's eyes can result in epileptic seizures. We need to make sure that the user's safety is not endangered by our product. We plan "to accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment".

Additional ethical questions arise when we look at the future of BCIs that can be based on our design. We hope to achieve a stimulator for a BCI that allows users to operate a computer using just their eyes. Commercial applications could make the world so easy to interact with that people would no longer need to move. If everything is able to be manipulated with the brain, people may become sedentary, which would result in a huge decrease in health of the general population.

We will also turn to many colleagues and teachers for help with our project. We will "seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others", most importantly, Ryan May, whose name will appear on all of our documents as TA.

3.4 Safety Considerations

One major safety concern for the stimulator is the risk of injuring the user's face, the area where the stimulator will be mounted on. The worst case scenario for this could be caused by the lithium battery exploding/expanding or an accidental short circuit in the PCB design which can cause the circuit to overheat. To prevent unwanted battery interactions with the face, we will design the mounting frame such that the battery will be placed in farthest away from the most sensitive

portions of the face most importantly the eyes. We will design the power circuit to include battery protection to detect when short circuiting occurs.

3.5 Contingency Plan

- If the PC interface does not work, the bluetooth cannot be demonstrated either. We will have to enter data manually into the Arduino using the Arduino IDE and simulate the rest of the circuit from there.
- If the bluetooth fails, we can connect the Arduino using a XBee Module which is bulkier and requires more work to transfer the serial properly but because there are many premade libraries for this module, we can still integrate the XBee as an option for our wireless module.
- If the Arduino proves incapable of clocking our LEDs at the desired frequencies accurately, we will attempt to build an external timing circuit to control the frequencies.
- If the TLC driver cannot control the intensity of the LEDs well enough using the Arduino PWM output pins, we may have to have to include several adjustable current limiting resistors.

4. Cost and Schedule

4.1. Cost Analysis

<i>Name</i>	<i>Hour Rate</i>	<i>Total Hour Invested</i>	<i>Total = Hourly Rate x 2.5 x Total Hours Invested</i>
Siyuan Wu	\$35	150	\$13,125
Bonnie Chen	\$35	150	\$13,125
Randy Lefkowitz	\$35	150	\$13,125
Total	\$105	450	\$39,375

<i>Item</i>	<i>Cost per Unit</i>	<i>Quantity</i>	<i>Total Cost</i>
LEDs	\$0.50	20	\$10
EGBT-046S Bluetooth SMD Module	\$7	1	\$7
7.4V Lithium Battery	\$16	1	\$16
Lithium Ion Battery Charger	\$39	1	\$39
Battery holder	\$1.29	1	\$1.29
Arduino UNO R3	\$39	1	\$39
Glasses Mounting Frame	\$60	1	\$60
PCB	\$25	2	\$25
TLC-5940	\$6	1	\$6
Total			\$203.29

Component	Cost
Parts	\$203.29
Labor	\$39,375
Grand Total	\$39,578.29

4.2. Schedule

<i>Week</i>	<i>Task</i>	<i>Responsibility</i>
2/05	Hand in proposal	Siyuan
	Order XBee transmitter and Wireless shield; start designing wireless module	Randy

	Order LEDs and microcontroller; write microcontroller code to control frequencies	Bonnie
2/12	Write mock design review	Siyuan
	Integrate wireless components into microcontroller code	Randy
	Find safe operating limit values for frequency of LEDs, integrate into Arduino code	Bonnie
2/19	Start writing DR, research on the BT module and order it, research on the LED driver.	Siyuan
	Design algorithm for the arduino program that decodes input from the PC	Randy
	Create code for integrating LEDs with wireless completed for DR as well as circuit diagrams	Bonnie
2/26	Finish Schematics and Verification	Siyuan
	Finish up Ethics and Safety and Verifications	Randy
	Finish up verifications, power and LED calculations for design review	Bonnie
3/05	Set-up connection between Bluetooth and Arduino, work on Arduino code with Randy if this is done.	Siyuan
	Create Schematic for Arduino to Serial Code	Randy
	Order and Design PCB on Eagle	Bonnie
3/12	Create Interface for LED frequency input	Siyuan
	Finish serial data code on Arduino	Randy
	Send in LED mounting frame for design	Bonnie
3/19	Write up Arduino code to take input and decode serial data	Siyuan

	Write up clock cycle/frequency output Arduino code	Randy
	Test LED Luminance levels with photodiode	Bonnie
3/26	Test and debug Arduino code that takes input and decode serial data	Siyuan
	Test and debug clock cycle/frequency output Arduino code	Randy
	Testing and design of power supply, Finalize Mock-Demo	Bonnie
4/02	Debug Integrated Arduino and PC code	Siyuan
	Debug Integrated Arduino and PC code	Randy
	Create draft of final report and presentations	Bonnie
4/09	Debug Integrated Arduino and PC Code	Siyuan
	Debug Integrated Arduino and PC Code	Randy
	Finalize mounting of all portable components	Bonnie
4/16	Integrate and Debug code with PCB board/Verify power dissipation in LCD driver matches proposed design	Siyuan
	Integrate and Debug code with PCB board/Verify correct output LED frequencies	Randy
	Verify power consumption and dissipation in circuit. Continue working on final report	Bonnie
4/23	Finish up Final Demo for Presentation	Siyuan
	Finish up Final Presentation	Randy
	Finalize Final Report	Bonnie
4/30	Final presentation and turn in final report	Siyuan

	Final presentation and turn in final report	Randy
	Final presentation and turn in final report	Bonnie

5. References

- [1] TLC-5940 Data Sheet. Texas Instruments
<<http://www.ti.com/lit/ds/symlink/tlc5940.pdf>>
- [2] Arduino Uno R3 Schematic. SparkFun Electronics
<http://arduino.cc/en/uploads/Main/Arduino_Uno_Rev3-schematic.pdf>
- [3] EGBT-045MS Bluetooth Module. Rasmicro
<<http://www.rasmicro.com/Bluetooth/EGBT-045MS-046S%20Bluetooth%20Module%20Manual%20rev%201r0.pdf>>
- [4] Luminous Efficacy Tables. Georgia State University
<<http://hyperphysics.phy-astr.gsu.edu/hbase/vision/efficacy.html>>
- [5] Photodiode Tutorial. Thorlabs
<http://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=285>
- [6] Guidance for the reduction of photosensitive epileptic seizures cause by television. ITU
<http://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.1702-0-200502-1!!PDF-E.pdf>
- [7] TLC5940 Arduino Library. Google Code
<<http://code.google.com/p/tlc5940arduino/>>