12/6/2013

OKLAHOMA STATE UNIVERSITY

ECEN 4013: TEAM 1 FINAL REPORT

ECEN4013: Senior Design | Team 1: The Cogs

Table of Contents

Title	ECEN4013: Team 1 Final Report
Author Names	Michael Bowman, Spencer Hood, Michael Lenth, Idwin Perente, Stephen Polczyski, Jason Semien, Jeff Szcinski II
Submission Date	December 6, 2013

Summary

- Describe what this product does- aim this description at the end user.
- Provide background information assuming the user does not know the context this product is used in.
- Describe briefly all inputs and outputs that are relevant to the end user (not a technical description)

Introduction

The Reflective Shield 2.0 was a capstone electrical engineering project in the college of Electrical and Computer Engineering. The Shield functions as both a weapon and armor for MAGE(Mobile Active Gaming Environment), which is implemented on the Oklahoma State University Stillwater campus. MAGE is an interactive gaming environment, which utilizes modern electronics to simulate medieval battles.

MAGE

In MAGE, each player is given a headband and an HIU. The two items are carried by the player and communicate with one another. The headband monitors the player's health and sends information to the primary server. The primary server holds the information of all players currently involved in MAGE. While playing in MAGE, players can equip a variety of weapons and armor. The HIU provides and interface for players to equip these items.

The Reflective Shield 2.0

The reflective shield 2.0 is a remarkable defense type of armor that the user attaches it to his/her left or right arm. Similarly to the oldest form of a shield, the Reflective Shield 2.0 serves the same purposes: a protective instrument designed to block attacks. In the MAGE system the Reflective Shield 2.0 brings an innovative feature that allows the player to use it for both offensive and defensive maneuvers when he/she is in the middle of the action. As the name suggests, it is a reflective shield that allows the player to attack enemies by reflecting a previous attack.

When the shield absorbs an attack, the energy is held by the shield for five seconds. During those five seconds the user can emit an attack or a healing spell with the simple press of a button. However, if the shield does not emit an attack or a healing spell when the five second countdown ends, the shield will take damage. After a certain amount of damage has been taken, the shield will break and can no longer absorb the magical attacks casted at the user.

The Reflective Shield 2.0 is outfitted with various indicators that notify the user both visually and audibly when the shield:

- absorbs energy from an attack,
- is running the five second timer,
- emits an attack,
- emits a healing spell,
- takes damage,
- is destroyed due to damage taken.

The front of the shield contains a large array of lights that create different graphics for the scenarios above. In addition to the front lighting, the shield also contains an audio system that plays different sounds for the scenarios listed above.

The shield has a large battery life that guarantees over two hours of use on the field of battle. The shield can be recharged when the battery is low. With a simple interface, and intuitive indicators, players in the MAGE battlefield can quickly put the Reflective Shield 2.0 to use.

Description (10 pts)

- Describe what this product does for a technical user short on time (be brief).
- Write a bulleted list of the products features
- Provide enough background information to put the project in a technical context.
- Describe briefly inputs and outputs of the product (a technical description but not tables of data).
- Provide a level 0 functional block diagram. This block diagram should be a physical layout of the product showing the physical location of connections to the device.

The Reflective Shield 2.0 is an infrared emitter and receiver for the MAGE system. Magical attacks are sent through infrared light at 56KHz. The infrared messages are communicated using a standard called MIRP (MAGE Infrared Protocol). Through this method of communication, the shield can send damage and healing “spells” to other players and object in MAGE.

The shield has a large array of individual LED’s and RGB LED’s that notify the user and his/her opponent when an event occurs (the shield turns on, the shield receives a valid infrared message, etc.). The shield also contains an exceptional PUI audio system, which is also used for notifications for event occurrences. The shield contains by a PIC24 model µController, which includes an onboard CAN Bus module. Through CAN Bus, the shield can communicate to the HIU as events occur.

Product Features

- Attack and Heal Interface
- Shield Damage Meter
- Fabulous design can be used on left or right hand
- Audio Notifications
- Lighting Notifications
- Infrared Receiver Capable
- Infrared Transmit Capable
- HIU Communication
- 2-Hour Rechargeable Battery Life

Technical Background

The Reflective Shield 2.0 is compatible with the MAGE system. In MAGE, each player is given a headband and an HIU. The two items are carried by the player and communicate with one another. The headband monitors the player's health and sends information to the primary server. The primary server holds the information of all players currently involved in MAGE. While playing in MAGE, players can equip a variety of weapons and armor. The HIU provides and interface for players to equip these items. All wearable MAGE equipment connects to the HIU through a USB cable that is provided.

Equipment held by the players communicates attack spells and healing spells through infrared radio waves. MAGE uses a standard called MIRP (MAGE Infrared Protocol Specifications), which defines the structure of the information sent through infrared.

Input/Outputs

The Reflective Shield 2.0 is outfitted with various indicators that notify the user both visually and audibly when the shield:

- is running the five second timer after absorbing an attack
- emits an attack
- emits a healing spell
- takes damage
- is destroyed due to damage taken

The front of the shield contains a light array that displays different graphics for the scenarios above. In addition to the front lighting, the shield also contains an audio system that plays different sounds in conjunction with the lighting displays.

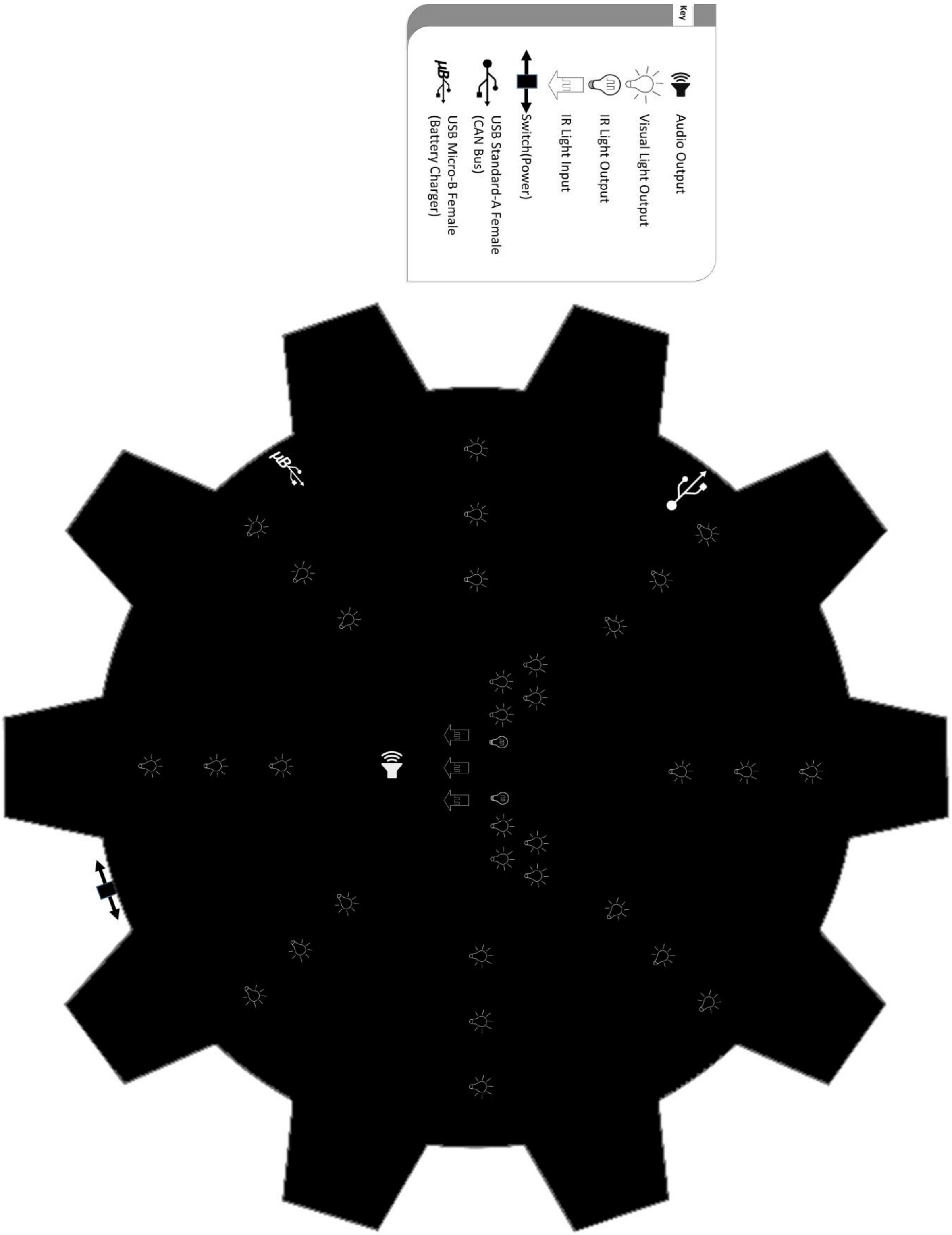
The Reflective Shield 2.0 must be able to communicate changes that happen on the battlefield. The HIU is able to broadcast occurrences wirelessly to the MAGE servers. The Shield is able to connect to the HIU and send and receive messages through CAN Bus.

The shield communicates with other weapons and accessories in the MAGE system through infrared light. The shield can receive incoming transmissions. The receivers can receive the signals from a 180º degree angle relative to the front face of the shield. The shield can also emit infrared light with a 90º spread to send magic attacks and healing spells to other players.

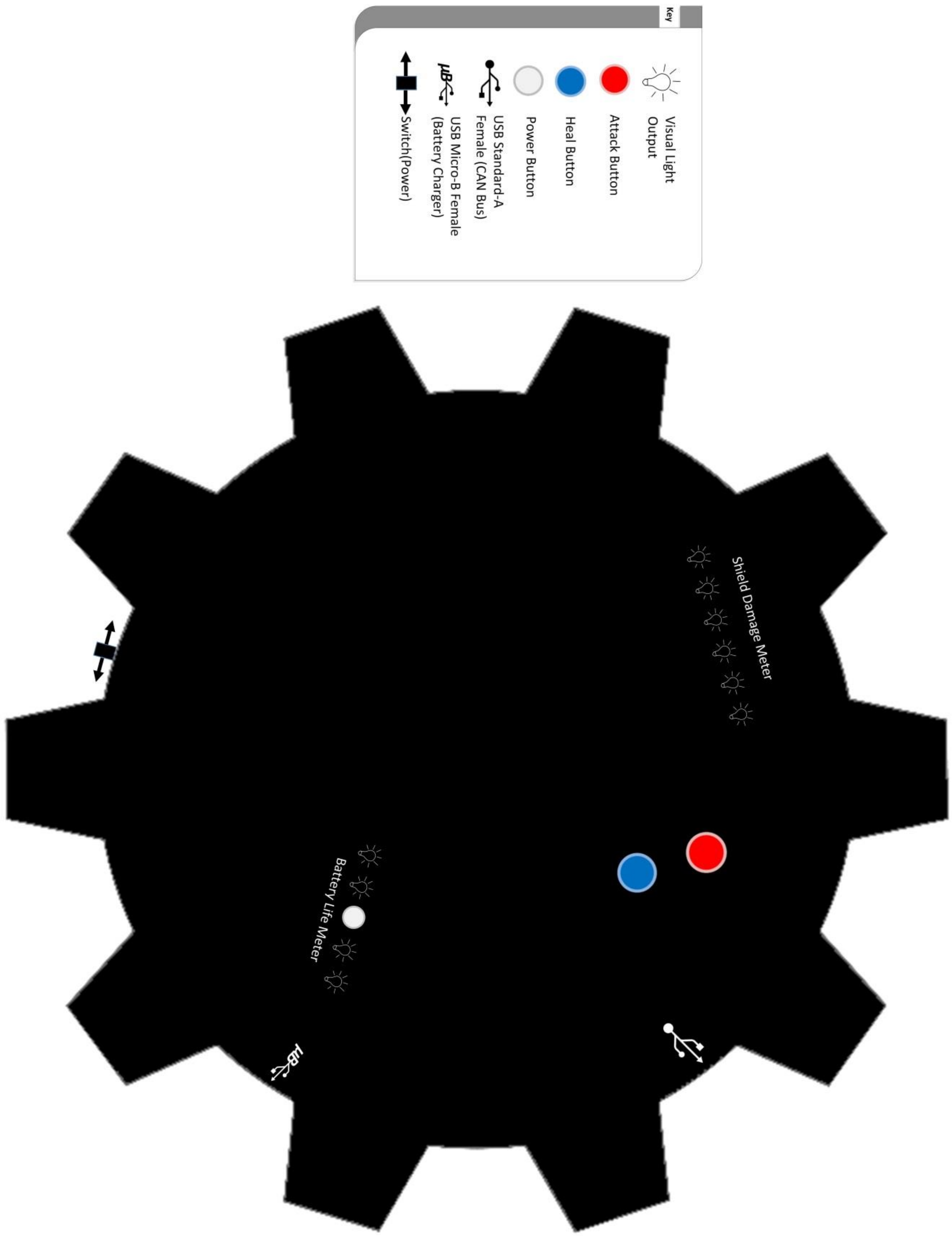
The back of shield will contain the main input controls for the user to operate with the shield. When an attack is received through the infrared sensors, the user will be able to select whether to attack and heal.

The Reflective Shield 2.0 can last at least two hours in the MAGE battlefield. The battery is also rechargeable and can be recharged with standard U.S power. Both the charging port and the on/off switch will be accessible on the external covering of the shield.

Level 0 Block Diagram (Part 1)



Level 0 Block Diagram (Part 2)

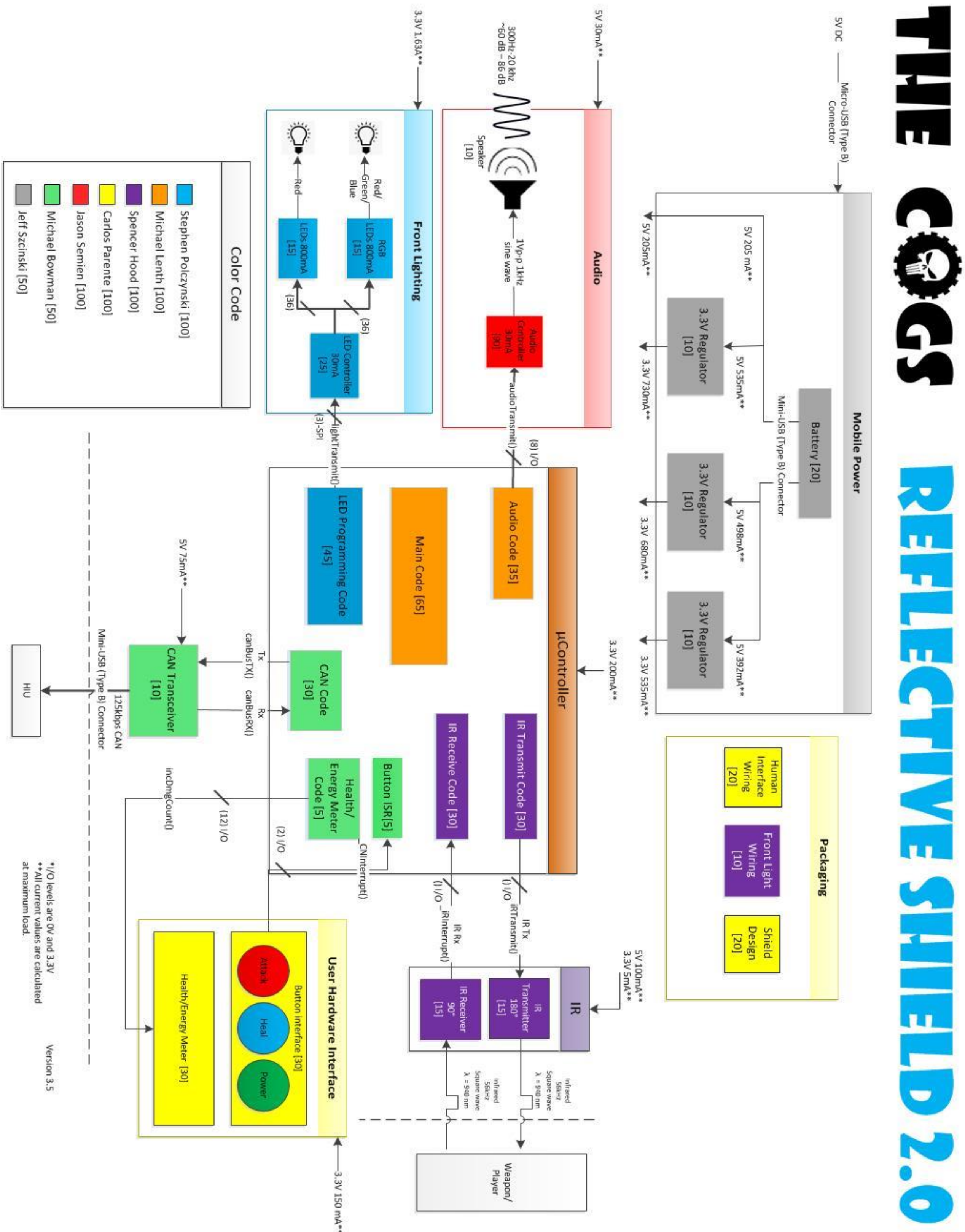


Product Block Diagram and Pin Out Diagram (10 pts)

- On one sheet of paper provide a level 1 functional block diagram of your product.
- On a separate sheet of paper provide a diagram table of the inputs and outputs for the level 1 block diagram as a system. If you are doing a project that has a significant high-level (e.g. C or C++) software component, include a table of public functions. (inputs and outputs overall: Considering the shield as whole)
  - ONLY INCLUDE INPUTS AND OUTPUTS OF THE SYSTEM FROM THE LEVEL 1 BLOCK DIAGRAM.
  - DO NOT INCLUDE INTERNAL CONNECTIONS/FUNCTIONS THAT ARE NOT ACCESSIBLE TO THE USER.
  - If your device has visual/tactile/audible user interfaces (such as displays, sounds, buttons) include these in the table.
  - Make sure you include detailed diagrams of any connectors with pin numbers.
  - The pin numbers and connector designations should match the Reference Designators from your layout and also the terminology used in your level 1 block diagram.

NOTE: No text on any drawing or schematic can be less than 10 point font, and the minimum component dimension on any drawing is ¼" (6 mm). Your block diagram must identify each part of your system and who was responsible for that part. Your block diagram may be on a separate page if required. If needed you are encouraged to insert a page larger than the standard letter size. To do this: 1) *Insert* → *Break* → *Page Break*. 2) *File* → *Page Setup*. In the dialog box select the *Paper* tab and set the size to custom. Then select *From this point forward*, from the *Apply to:* pull-down menu. 3) Select the paper size you need for your drawing. Sizes up to 22"×22" are supported. 4) Insert another page break. Follow the instructions in step 2 to set the rest of the document back to letter size (from the new page break forward)

Level 1 Block Diagram



Input/Outputs

Category	Name	Descpription	Interface
Power	5V DC	The Anker® 3E battery requires USB power in order to recharge	A Micro-USB B Male connector is required to charge the shield. (**figure 1**)
Human Interface	Power Button	This button is located on the battery. It will power up the battery.	N/A
Human Interface	Switch	This switch is located on the bottom of the shield. This switch connect the Shield electronics to the battery when set to the "on" position.	N/A
CAN	CAN (62.5kHz)	This allows the HIU to interface to the Shield for CAN communication.	A standard USB type-A to Mini USB type-B Cable is required to connect the Shield to the HIU. (**figure2**)
Audio	Speaker	Outputs 6 sounds that are audible indicators for the functionality of the shield.	N/A
Infrared	IR Transmit	Transmits infrared light (56 KHz)	N/A
Infrared	IR Receive	Receives infrared light (56 KHz)	N/A
Front lighting	RGB	Red 630 nm at 1300 mcd, green 525 nm at 5000 mcd, blue 472 nm at 1000 mcd all at 15 degree view angle. Also indicators for functionality	N/A
Front lighting	Red LEDs	Red 625 nm at 11000 mcd and 20 dgree viewing angle. Also indicator for functionality	N/A
Human Interface	Push Button	The buttons are located on the back of the shield side of the damage meter. It gives the command to Heal or Attack.	N/A
Human Interface	SMT LED	The damage meter- Red, Yellow and Green SMT LEDs are located on the back of the shield as damage indicators.	N/A

Table 1: Inputs/Outputs of Reflective Shield 2.0 System

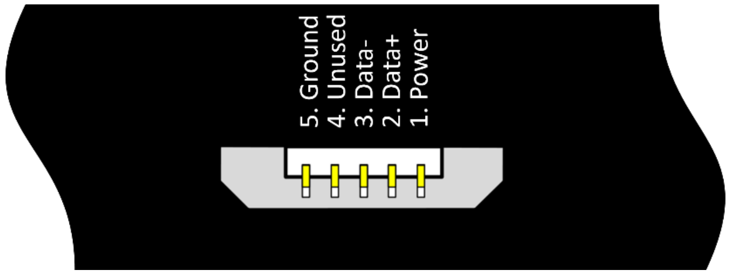


Figure 1: Battery Charger Port Pinout (USB Micro-B)[1]

Figure 2: CAN Bus Port Pinout (USB Standard-A)

[1] http://www.kineteka.com/microusb-b.aspx

Detailed Technical Description (15 pts)

- Describe the principles of operation of your device in detail.
- Provide a brief description of the signals/data at all inputs, outputs and internal test points.
- Describe the test procedures used to verify proper device operation.
- Provide a brief description of debugging techniques and common problems the user might encounter with your device.

Voltage Regulators

Regardless of the power source chosen, there are usually differing voltage requirements for each function block of a design. Providing the correct consistent voltage and current necessary is crucial for proper functionality.

Switching Regulators

Switching regulation is more complex compared to other regulation techniques (such as linear regulation). But with this complexity allows for higher efficiency and usually higher current.

The power of an electronic component depends on the current and the voltage.

$$P = IV$$

Switching regulators reduce voltage, while keeping the power delivery relatively stable. Since there is a decrease in voltage, more current can flow. A switching regulator that regulates from 5V to 3.3V has the potential to supply 151.5% of the input current. The current output will not be that high in reality, but switching regulation does allow for lower current input while meeting the current output requirements.

The Reflective Shield 2.0 is equipped with three Recom R78B1.5-1.0 switching voltage regulators[2]. The maximum current drawn by the shield is 2.15 A. With the added efficiency of the regulators (assumed to be 90%), the maximum current drawn from the power supply is only 1.63 A.

In addition to the power efficiency, the R78B1.5-1.0 regulators also have a simple three pin layout. The R78B1.5-1.0 was designed to be a replacement for linear regulators.

---

[2] Recom

Power Supply

The Anker Astro 3E[3] battery is the central power supply for the Reflective Shield 2.0. The Astro 3E has a ten-thousand mAh capacity as well as two outputs that can provide 1.5A each. The Battery interfaces with the motherboard through two Mini-USB Type-B connections. The battery has its own internal regulation and provides 5V with little ripple. If at least 45 mA is drawn from the battery, the Astro 3E will continue to stay on until depletion.

The Astro 3E has a built-in battery life meter and power button. The battery life meter composed of four blue lights. As the battery life decreases, the lights will decrease. The power button provided on the Astro 3E only turns on the battery. When pressed, the battery life meter will illuminate. If the power switch on the underside of the shield is switched to the “on” position, the shield will turn on and the lights on the front face of the Reflective Shield 2.0 will illuminate and the initialize audio will play. However, once the shield is “destroyed” it does not automatically power down. To power down the Reflective Shield 2.0 the power switch must be switched to the off position. The battery life meter will remain on for approximately 15 seconds.

µController

The µController is the “brain” of the system. It provides the ability to interact with all of the other components of the shield. This is done in a multitude of ways. The digital I/O pins interface with audio, the pushbuttons, infrared transmit, and the damage meter lighting. The SPI module on the µController is used to communicate the front lighting information to the daisy chained TLC5952 chips. The PIC has a built-in CAN module which communicates with the MAGE HIU through the CAN communication protocol. And the input capture modules located on board are used to register the widths of the infrared pulses as received by the infrared receiver. The µController used for the Reflective Shield 2.0 reflected the needs of the system’s components as previously listed.

The µController chosen was the PIC24HJ128GP504. This µController runs at a high instruction rate (40 MIPS), which was necessary for updating the front lighting. Twenty-

[3] Anker

six pins are required by the system components and two pins are dedicated to power the chip, requiring at least twenty-eight pins total. Therefore, the PIC24HJ128GP504 was chosen due to the high operating speed, availability of CAN/SPI modules, and 44 available pins, leaving the possibility for design changes if necessary.

If looking at the microcontroller with Pin 1 in the bottom right corner (reference **Figure** 3), the left side is wholly dedicated to audio output. These pins are set up as outputs and set high (active state is low).

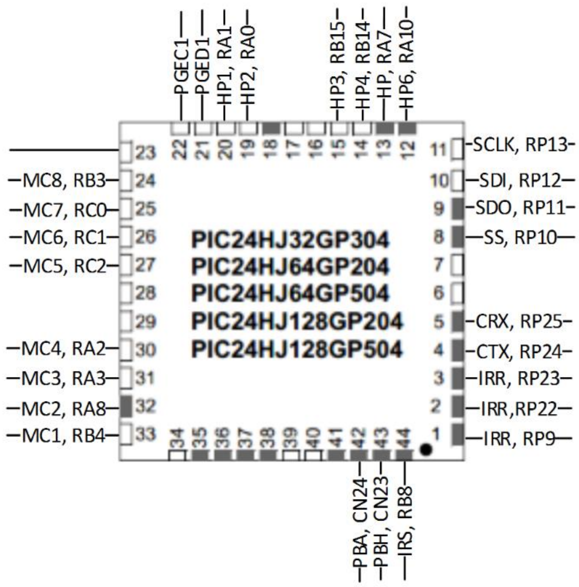


Figure 3: µController Pinout[4]

[4] M2

Audio – M1-M8; 8x I/O Pins (8x Output)
SPI (Lighting) – SCKL, SDO, LAT; 2x Re-mappable Pins, 1x I/O Pin (1x Output)
CAN – CRX, CTX; 2x Re-mappable Pins; (1x Output, 1x Input)
Push Buttons – PBA, PBH; 2x CN Pins (2x Inputs)
HP LEDs – HP1-HP6; 6x I/O Pins (6x Outputs)
IR TX, IR Rx – 3x IRR, 1x IRS; 3x Re-mappable Pins, 1x I/O (3x Input, 1x Output)
Programming – PGEC, PGED ; 2x Programming Pins

Table 2: µController Pinout Detailed Description

All damage meter LED's are set as outputs and initially turned on to indicate full health. Three pins are used for SPI: one I/O for latch and two re-mappable pins for the clock and output data, RP10 and RP11, respectively. The SPI module is set up by enabling the clock, setting the data output to be controlled internally, enabling word-wide communication, setting the clock idle state to idle on low, enabling the master mode, and enabling the SPI module and interrupts. The complete set-up for CAN communication will be discussed in detail in the CAN section of the report. The two are for reception and transmission, respectfully. The next component set up was the infrared transmission and reception. Transmission simply utilized an I/O pin on the bottom of the microcontroller. Setting up infrared reception was more complex as two captures modules were needed. Interrupt capture 1 (IC1) and interrupt capture 2 (IC2) were both used to "capture" the rising and falling edges of the magical attack packet. The timer used to get the times of each edges was Timer2. IC1 was set to interrupt on every capture event triggered by a rising edge and IC2 was set up to interrupt on every capture event triggered by a falling edge. The priorities of these interrupts are set to the highest possible, priority 7. The pins dedicated to the interrupt captures are on the bottom right side of the microcontroller. The last two things which needed initialization were the push buttons and the countdown timer. The pushbuttons pins are located on the bottom of the microcontroller and are set to be input pins. The will however use a change notification interrupt. Therefore, the change notification function on each pin was activated and the single interrupt vector that services the change notifications for all the pins was also enabled. The countdown timer was assigned to Timer 4. This timer derives its rate from the internal clock and uses a 1:1 pre-scaler. The interrupt that services the timer 4 is set to priority level 3.

Infrared

The shield receives infrared light through three receivers. Each receiver is capable of receiving signals within a 90º range. The three receivers are angled such that infrared can be received from 180º. The receivers have an internal carrier frequency of 56 kHz and actively suppress signals of other frequencies. The receivers are active low devices and only output a low signal when they are actively receiving data. This data is sent to the µController. The µController incorporates two interrupts, which occur on every falling and rising edge. The length of the transmission is calculated and compared to the known lengths of the damage packet. If the transmission length is correct, the shield counts and responds to the transmission.

The shield sends out infrared light through two infrared light emitting diodes. These diodes transmit infrared light at 940 nm and over a range of 90 º. The diodes are capable of transmitting valid signals over a distance of 35ft with less than a 10 percent error rate. This is accomplished through the use of high current through the diodes. To accommodate this high current a bipolar junction transistor has been implemented. This gives the pic microcontroller precise control over the current through the diodes. The pic sends out sequential busts of 56KHz square wave signal to the base of the transistor, which results in bursts of infrared light from the diodes.

Audio

The reflective shield required audible indications when the shield received an IR magical attack, followed by a countdown sound to indicate the user they have 5 seconds to cast back the magical attack or cast a healing IR packet. The shield also required audible indication for when the shield broke after being depleting all of it health points.

All the sounds were stored on the ISD1932 integrated circuit chip, which is a low active chip that can store up to 8 different audio sounds of certain lengths. The duration of the sounds are designated by an external resistor value. In the reflective shield a 100kΩ resistor to allow for a 5 seconds to be allocated for each of the 8 memory location on the ISD1932. The ISD1932 chip acted as full audio system including memory, record, playback, and amplification. To save the sounds into memory locations required pre-hardware programming by setting a pin on the chip into record mode. Once in record mode the sounds can be saved on the chip by an analog input mono auxiliary cable. Once the IDS1932 is programmed with the 6 sound used by the shield, it communicates with the microcontroller through GPI/O. When a certain action takes place by the shield the microcontroller sets the certain pin with the correct sound low to 0V causing the stored sound to be played back when the ISD1932 is set in playback mode. The chip operates at 5V to allow for the maximum volume of the output sound. However the 8 message pins connected to the microcontroller have a 3.3V drop across them, because the microcontroller runs on 3.3V. The ISD1932 plays back through a PUI 8 ohm speaker that has an 86dB output power level. The PUI speaker outputs the 6 sounds with a frequency range between 240Hz to 18 kHz. With the ISD1932, PUI speaker, and a few other smaller components the audio for the shield were complete.

CAN Bus

CAN Bus on the Shield is a serial-communications broadcast network. It is capable of sending and receiving messages at 62.5 kbps. The Shield and HIU are interfaced with a standard USB type-A male to mini-USB type-B male cable. The Shield is protected from any accidental transient voltage spikes that might occur when connecting the mini-USB cable via a set of 24V varistors between each of the lines going to the cable.

The Shield can read and write data onto the CAN bus via the use of Direct Memory Access Random Access Memory on the microcontroller. By utilizing DMA RAM, data is written and read into memory without CPU intervention. Whenever the read buffers are full, a CAN receive interrupt is thrown by the DMA module, signaling the CPU that a message has been received and can be processed by the CPU. Whenever data is ready to be transmitted onto the CAN bus, a transmission request interrupt flag is set to let the DMA module that transmission needs to occur and it will write all the data out onto the bus. Data transmission and reception is done utilizing the FIFO buffers in DMA RAM.

The microcontroller (PIC24HJ128GP504) only writes serial data in and out of its ports. It is split into the CANH and CANL lines by the MCP2551 CAN Transceiver. This integrated circuit is able to convert the signal seen on the TX line into the differential voltage required by the Bosch CAN protocol. It also converts the CANL and CANH signals seen into a normal serial signal that is seen on the RX line. The MCP2551 is capable of running at frequencies up to 1 MHz, which are common in CAN 2.0B systems.

Battery Troubleshooting

The battery life meter blinks when the battery power button is pressed.

1. Make sure the battery is fully charged (refer to the "Charging the Battery" section on page ___________).

The battery life meter is on but the shield is not on.

1. Turn the power switch to the "on" position.
   - If the Reflective Shield 2.0 is still not turning on proceed to the next step.
2. Open the shield casing.
3. Check that the two standard USB A-Type connectors are plugged in to the Anker Astro 3E battery outputs.
4. Check that the two Mini-USB B connectors are securely plugged into the ports on the motherboard.
   - At this point refer to the voltage regulation troubleshooting section (the next section)

Voltage Regulation Troubleshooting

The voltage regulation troubleshooting steps require the following equipment:

- Voltage Meter
- Needle Probes

> A static discharge to the motherboard can cause permanent damage. Make sure all static is discharged by touching your bare hand to a piece of metal.

1. Put the ground probe on the ground pin.
2. Put the voltage probe on the header pin(**figure ___________**).

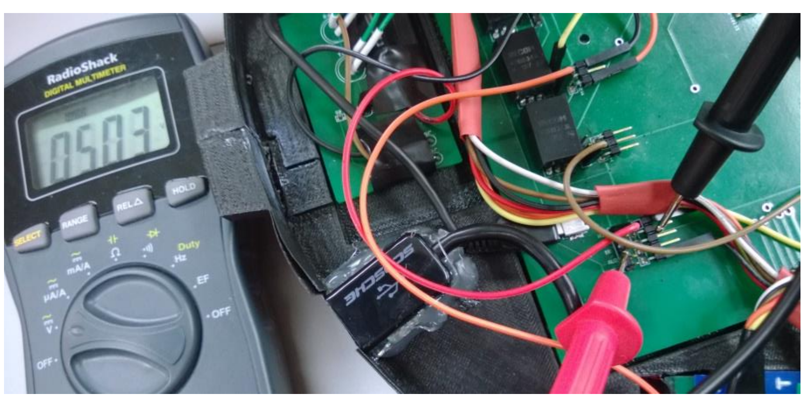


Figure 4: Step 1 and 2 Probe Demonstration

3. The volt meter should read approximately 5V.
4. Repeat step 2 but with the other three sets of power header pins (f**igure 5**). The volt meter should read approximately 3.3V for these pins.

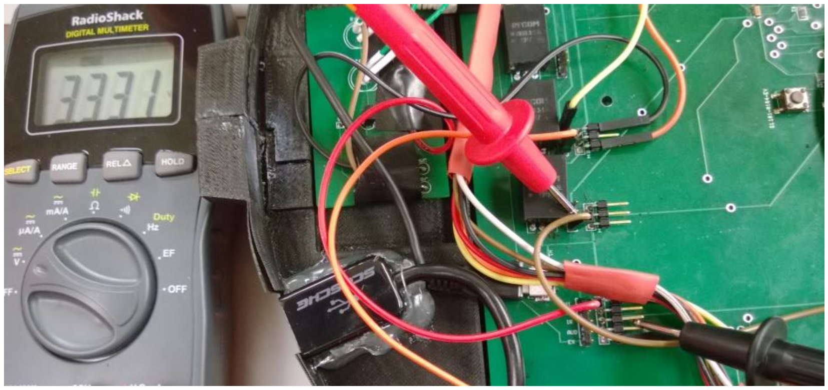


Figure 5: Step 4 Probe Demonstration

If any of the readings are incorrect the voltage regulators and/or the battery should be replaced.

Detailed Application Information (15 pts)

This section provides instructions to users on how to get your device working (in one or two paragraphs).

- Describe in detail how to use the device
- Describe in detail how to connect the device to other systems
- Describe common fault conditions and how to correct them
- Describe any variable voltages, resistances, currents, or any other user selectable hardware features
- Describe if necessary how to calibrate the device and or test for proper operation.
- Describe how to program any onboard memory or microprocessor

Turning on the Shield

Move the power switch located on the bottom of the shield to the "on" position. Push the power button located on the back of the shield is provided which turns on the battery. Four blue lights, which are visible on the back of the shield on either side of the power button, indicate the status of the battery. As the battery life decreases, the number of blue indication lights will also decrease (**figure 6**). If the battery is dead, the four blue indicator lights will blink several times when the power button is pressed (refer to the "Charging the Battery" section for instructions on recharging the battery).

Figure 6: Battery Life Indicators

After the shield is turned on, the shield will play its power up sound the LEDs will light up in a circular pattern around the shield and then turn off. Connect the USB-A to Mini-B cable to the HIU and the shield's USB-A port. The shield will continuously broadcast a connection request to the port until it has been acknowledged by the HIU. After this happens, the RGB LEDs in the skull's eyes on the front of the shield will turn green. Now, the game is ready to be played.

Connecting to the Server

Plug in the Mage Headband (sold separately) to the USB port on the top right side of the shield. The eyes of the skull will activate and emit a green light. The shield is now initialized with the mage server and is ready for operation in the game.

Programming the Microcontroller

A PICkit 3 is used to program the µController. There are six pins (1-6) on the device that indicate the required connection to the microcontroller when programming (**figure 7**).

- Pin 1: MCLR pin
- Pin 2: Vdd
- Pin 3: ground
- Pin 4: PGD (on some PIC's, PGED)
- Pin 5: PGC(on some PIC's, PGEC)
- Pin 6: Not Connected

It should also be noted that a resistor between 4.7K and 10K ohms. AV_{dd} and AV_{ss} should be connected to power and ground as well. Once these connections are all made, the user can begin programming the microcontroller using MPLAB X. Upon opening the program and setting up the desired microcontroller, PICkit 3, and programming language, the user can begin work on the code. After the code is written and ready to be programmed, the user should click the "PROGRAM" button. This will compile the program and abort programming if any errors occur within the code. Upon successful compilation, the words "Program Successful" and the microcontroller will then be programmed.

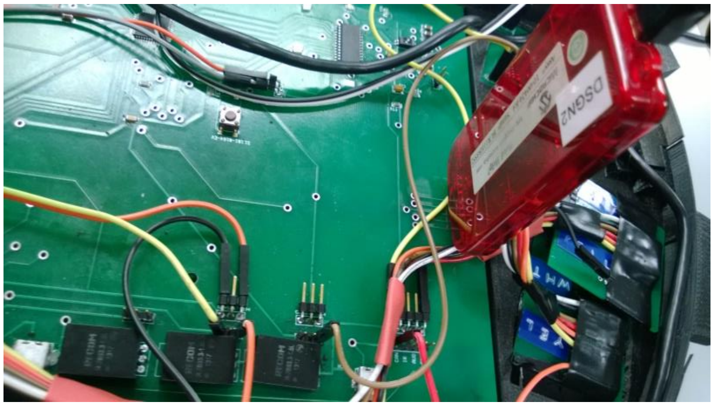


Figure 7: PICkit 3 Connection[5]

---

[5] M1

Application Example (10 pts)

- Describe in detail an example of how the device would be used by the end user.
- Provide a connection diagram specifying needed hardware, cables, power supplies, etc.

Playing the game

The shield waits in this stage waiting to receive infrared input from another player. When the shield receives a valid damage attack from another player it gives a light and audio indication. This event triggers a five second countdown to begin. The player has five seconds to decide whether to send the damage packet back, by pressing the red button, or to convert it to a healing packet and send it to a teammate, by pressing the blue button. If no choice is made by the user the shield will take the damage itself and decrease its health by one, indicated on the led display on the back of the device. The player should run around the playing field absorbing attacks from the enemy and either fire them back, or heal the players' allies.

Breaking the shield

If the user of the shield fails to make a decision 6 times the shield will break and emit a lighting and audio routine. The shield will then turn off all of the leds and sit in an idle state until the end of the match. The shield can be reset manually by turning the power off and back on again, but doing so midgame is cheating and not tolerated by the NMC (National Mage Committee).

Damage Reception

The shield is capable of receiving damage MIRP packets from other devices. Upon reception of a MIRP damage packet, the shield's five second countdown timer will begin with a short attack received LED routine on the front of the shield. The user will be audibly notified of the 1^{st}, 3^{rd}, and last second of the countdown. The LEDs on the front of the shield are organized into three rings around the shield. Each second will cause a different ring to light up, starting on the outside, then moving to the middle, then the inner ring, then to the middle again, and finally to the outside.

Reacting to Damage

After the countdown routine has begun, the countdown can be interrupted by pressing one of the blue or red buttons on the back of the shield. By pressing the blue button, the shield will use the energy captured by the attack received to heal an ally, sending the energy back as a MIRP heal packet. If the red button is pressed, then the energy will be used to reflect the damage packet away from the shield, potentially damaging another player or weapon. Lastly, if no button is pressed, the energy that was stored on the shield will damage the shield after 5 seconds, removing health from its hit points. This can be seen visually with the HP meter on the back of the shield. After the shield has taken 6 LEDs worth of damage, the shield will break and no longer be able to be used during this game and the shield will broadcast to the server that the shield has been destroyed.

Charging the Battery

Many electronics, the Reflective Shield 2.0 included, use USB power (DC 5 volts). Standard residential power in the U.S. is AC 120V. To convert from AC 120V to DC 5V, an adapter is needed. These adapters will typically have a 120V AC plug and a female USB A-Type port.

The shield battery can be charged with a micro-USB B male connector. The cable must receive standard USB power. This is commonly accomplished using a USB power AC adapter (**Figure 8**).

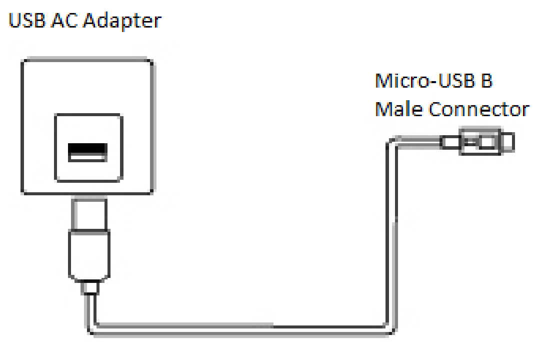


Figure 8: Typical Use of a USB AC Power Adapter[6]

> While the Shield's battery is fully compatible with all AC adapters, a USB AC Adapter with 1.5A or more output current will maximize charging speed. Lower current output will prolong charging time. ACK: Anker"

[6] Anker: Astro 3E External Battery

Team Organization and Management (10 pts)

- State how your team was organized and what role each team member played. Describe the ground rules set up for meetings.
- Describe how your team determined whether work was performed to a professional standard and how the team addressed issues of individuals who did not meet expectations.
- Identify specific and practical suggested improvements in team organization and management.
- Identify any specific weaknesses of your approach/performance and how you would fix them.

Team Organization and Roles

Michael Bowman was nominated as the Lead Engineer (LE) for our team in our first meeting, because of his prior knowledge about microcontrollers along with his other electrical and computer engineering experience. Beyond the standards an LE had to perform for the course, Michael Bowman was responsible for the hardware and software portion of CANbus. He was required to research and learn how to implement CANbus into our project along with communicating that information to entire team. He communicated this information in our Sunday morning meetings remotely by Skype do to the fact he commuted from Tulsa. In the final integration phase of the project Michael Bowman was in charge of fully integrating all of the sections in our shield.

Jeff Szcinski was nominated as the teams Project Manager (PM) in our first meeting as well. His background of leading previous technically projects, along with his experience with Microsoft Office contributed to him being nominated as PM by the team. Jeff used these skills to establish great organizational and communication methods between the team. Jeff was in charge of not just leading our team to complete the shield, but had the task of researching and supplying the entire shields power to operate on. He was responsible for find a sufficient power supply and then designing a circuit using regulators to supply 5V and 3.3V to different sections. The also required him to communicate with everyone section of the shield to make sure the available voltage was supplied to them.

Carlos Idwin was one of the hardware engineers responsible the Human Interface spectrum of the shield. Carlos has the task of completing the circuit boards for health indication and buttons for the user to interact and function the shield. He had to come up with a design to display the health of the shield, using LEDs and another design for how the buttons need to be positioned for the user. Along with this Carlos was responsible for creating a case design for the shield. Using case designs created by members of team, which were voted on by the team. Carlos had to take that design and create it in AutoCAD, for 3D printing. He was in charge for the location of all the components and boards with in the shield. In the final integration Carlos design was 3D printed and incased everything that makes up the shield.

Michael Lenth was one of the software engineers. He was given the task of researching the microcontroller and learning how to write in C and use MPLAB X. He then was given the task of passing on this knowledge to the rest of the group and also assist others in programming the microcontroller. He also worked with everyone to integrate their blocks of code into on signal program. Michael Lenth and Michael Bowman were the main software engineers who worked together to make the shield function properly. During the team meetings, Michael made sure the meetings stayed on task and within the time limits set on the agenda.

Spencer Hood was a hardware and software engineer. His task was to research and design the IR transmit and receive portion of the project. Spencer's role consisted of finding transmitter and receivers that could be functional in the MAGE Infrared Protocol. Also along with designing the hardware circuit boards for transmit and receive he had to use a microcontroller to program these boards to send IR signals required for the rest of the shield to function. Then this code was integrated by him and Michael Lenth to work with the main code.

Jason Semien was a hardware engineer in charge of audio. His responsibility was to research and design an audio circuit with the ISD1932 chip. He was in charge of make sure all the audio were programmed on the chip and the chip was functional correctly with the microcontroller. He had to also create and find the sound for the six audio indicators. Also he was responsible for selecting the PUI speaker that would provide the best audio quality and amplification. The audio system was put together by Jason and the code to call the audio was finished by Michael Lenth.

Stephen Polczynski was a software engineer in charge of lighting. He was responsible for the programing of the front lighting. Stephen had to research how to program a microcontroller in making the light communicate through SPI. He also had to use that knowledge to create different lighting displays to indicate to the user and others when the shield was performing a certain action. He also had to implement his code in the main code and make sure it performed correctly. He also designed the light by using two IC ship, RGBs, and Red LEDs to disable these different light states he coded.

Team Requirements and Performance

The team was required to meeting every Sunday at 10:00 a.m. in Engineering South 401 to conduct meetings. The meetings were formatted to discuss certain issues during the project as well communicate updates from everyone's reasonability. These meeting lasted about one hour to one hour and a half.

During these meetings lead by Jeff were used to deal with working professional and holding people accountable for the expectations that needed to be met. This was a way we delete with any team member falling short of their expectations. It was brought up to the entire team and corrected during these meetings.

Also the meetings pointed out certain areas of improvement the team could use. Some the areas being communication and team members meet deadlines. Also the team had certain medians to organize and communicate as a team about the project. These medians being Dropbox.com and GroupMe. Dropbox.com allowed us to organize all of documents and work completed for the project. GroupMe allowed for all of us to be in contact with each other through group texting. Dropbox.com and GroupMe along with spending time outside of working in the lab together on this project allowed for these issues to be resolved beyond the Sunday meetings. Overall the team came together as one and used teamwork to make our dream a fully functional shield work.

References (5 pts)

Use the IEEE Style. Details can be found on the course web site or in the IEEE Authors Manual. References must be cited in the body of the text.

[M1] – Microchip Technology Inc. "PICkit™ 3 Programmer/Debugger User's Guide," Internet: http://ww1.microchip.com/downloads/en/DeviceDoc/PICkit_3_User_Guide_51795A.pdf [Nov. 5, 2013].

[M2] – Microchip Technology Inc. "PIC24HJ32GP302/304,PIC24HJ64GPX02/X04 and PIC24HJ128GPX02/X04," Internet: http://ww1.microchip.com/downloads/en/DeviceDoc/70293G.pdf [Nov. 5, 2013].

[J1 -]Nuvoton Electronics Corporation. (2009, January, 5). ISD1900 Single-Chip, Multiple-Message Voice Record/Playback Device (Revision 0.51) [Online]. https://www.sparkfun.com/datasheets/BreakoutBoards/BOB-09579-ISD1900.pdf

Ian Lang Electronics. (2012, March). How to Work the Sparkfun Voice Recorder ISD1932 Breakout Board in Direct Mode [Online]. http://ianlangelectronic.webeden.co.uk/#/voice-recorder/4562321245

Anker (2013) Anker Astro 3E 10000mAh External Battery (Black) [Online] Available: http://www.ianker.com/product/79ANS1052-BA

Thermibel S.A., "What is infra-red?" [Online] Available: http://www.thermibel.be/documents/laboratory/labo-infrared.xml?lang=en

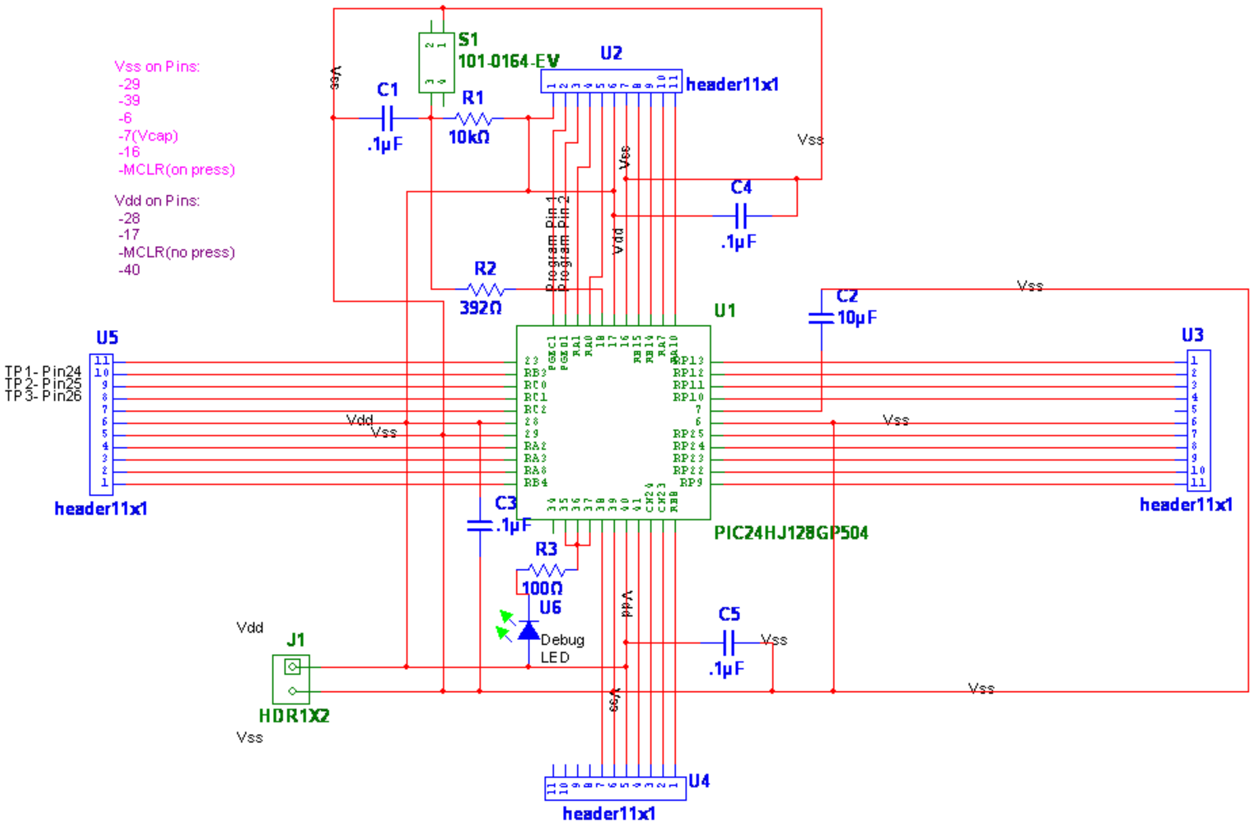
stack exchange inc (2013)For the CAN bus, is it OK to swap CANH and CANL lines? [Online] Available: http://electronics.stackexchange.com/questions/73303/for-the-can-bus-is-it-ok-to-swap-canh-and-canl-lines

Perfecto(2002-2011) "what is LED ?" [Online] Available: http://www.ledscreenchina.com/led-display-news/page/2/

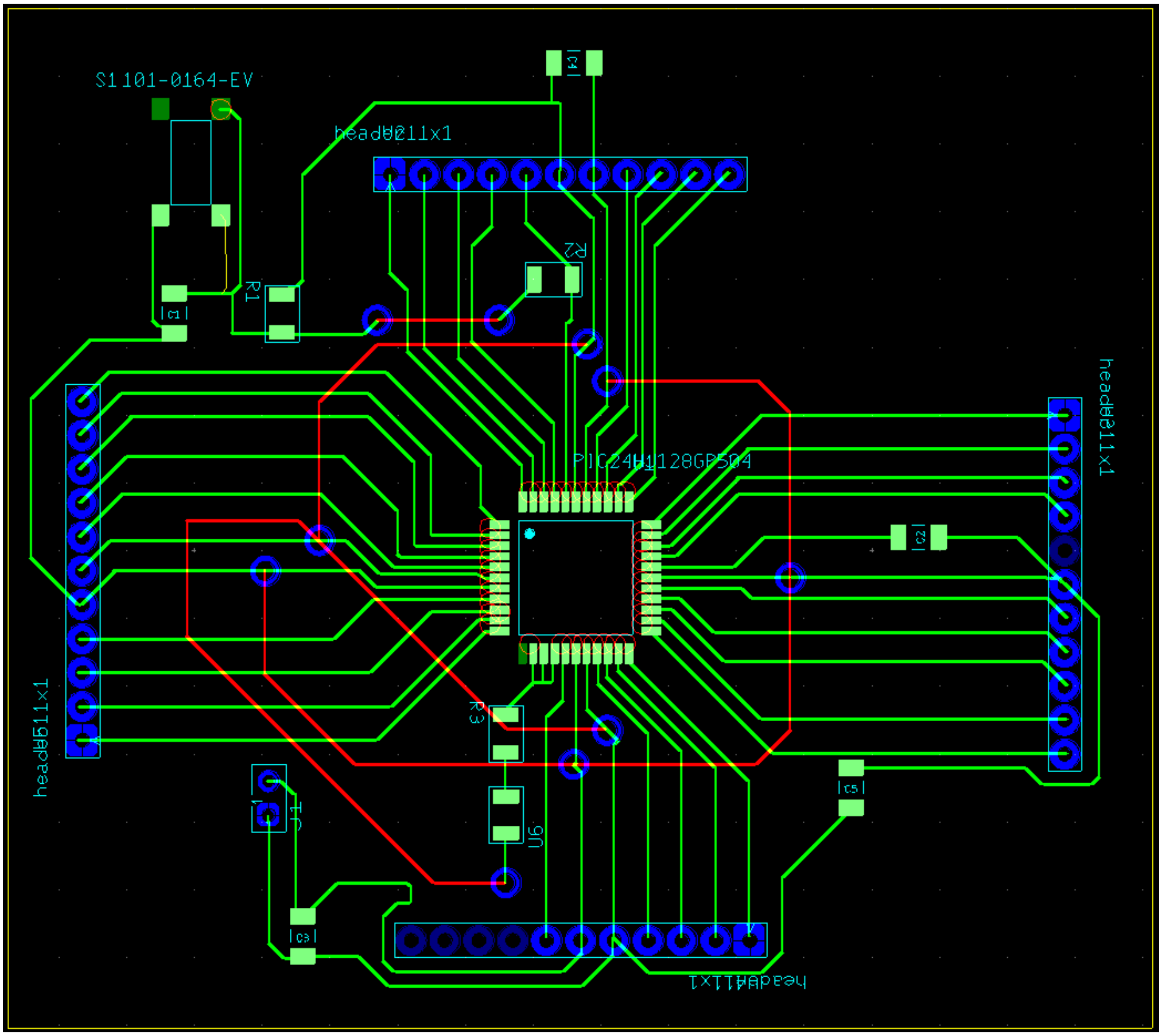
Texas Instruments(2013) "24-Channel, Constant-Current LED Driver withGlobal Brightness Control and LED Open-Short Detection" [Online] Available: http://www.ti.com/lit/ds/sbvs129/sbvs129.pdf

Appendix A (Schematics and PCB Layouts)(5 pts)

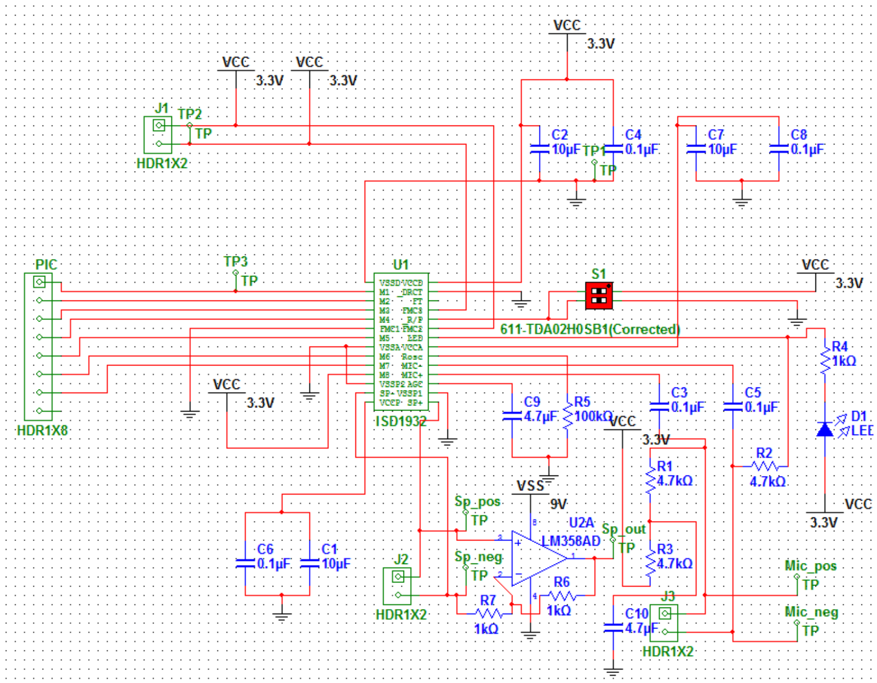
- Provide any relevant diagrams, schematics, PCB Layouts, Code, or other technical documentation.



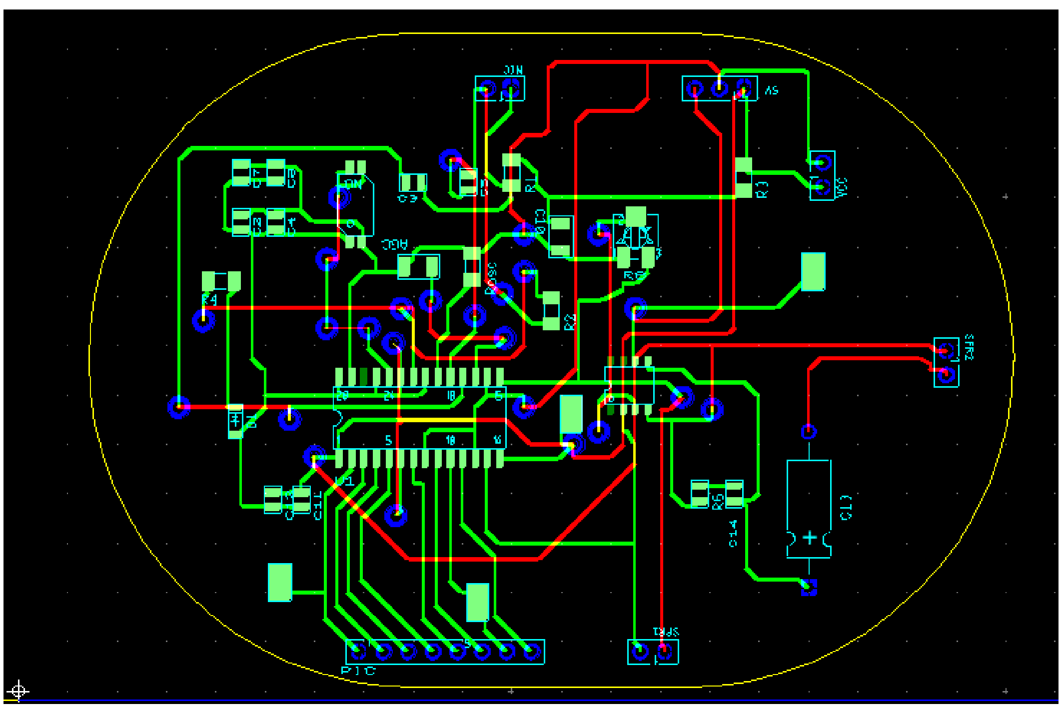
Microcontroller Schematic



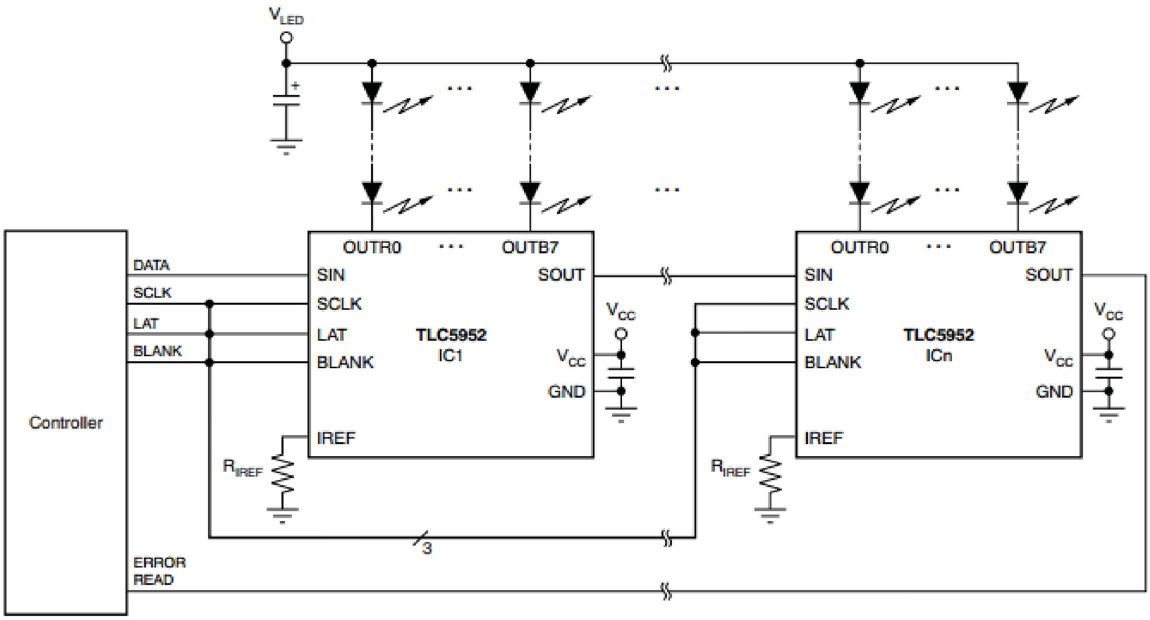
Microcontroller PCB Layout



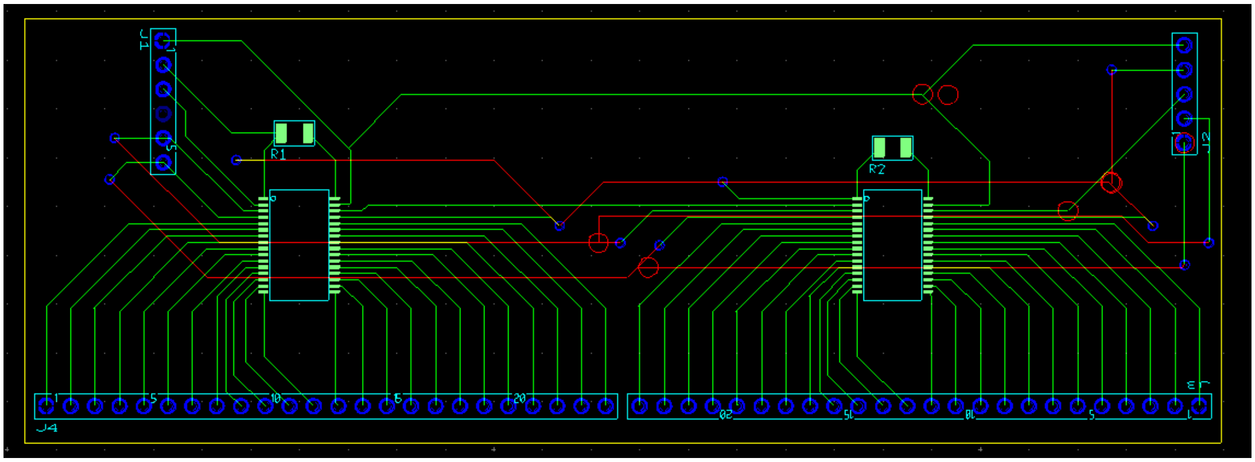
Audio Schematic



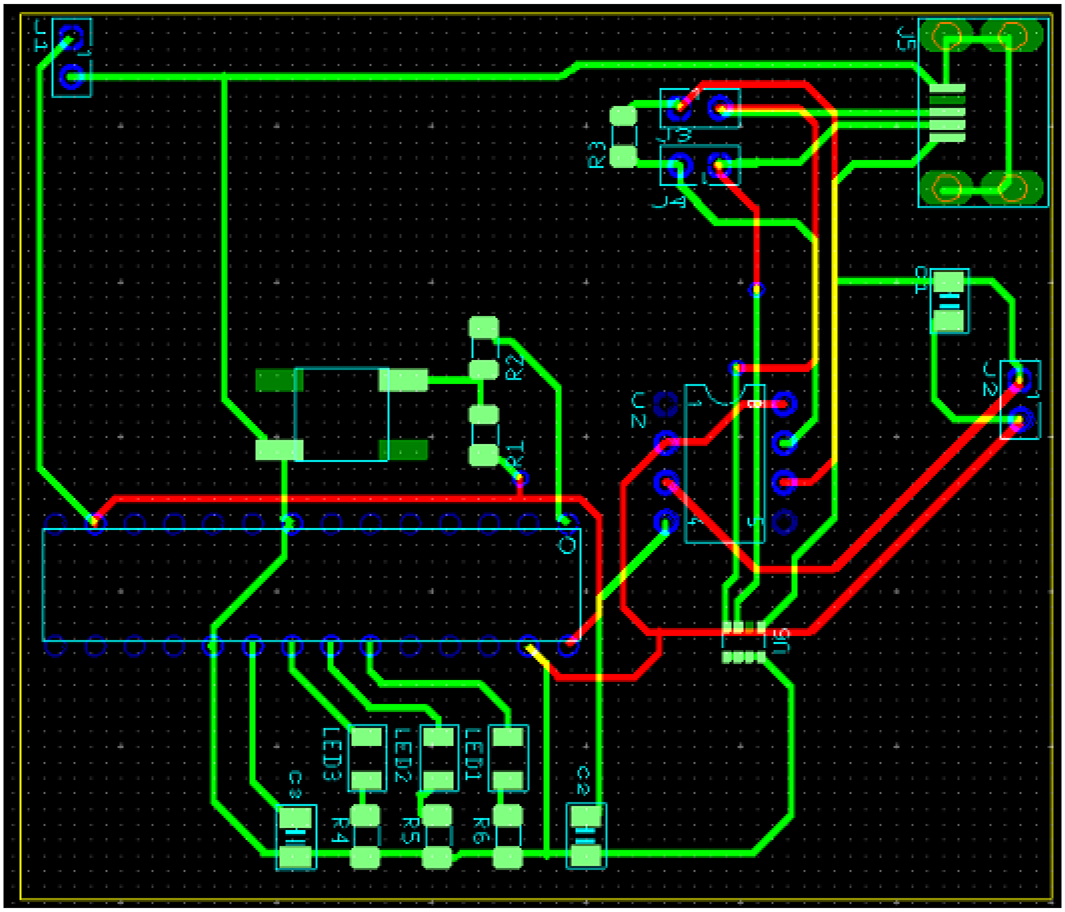
Audio PCB Layout



Lighting Schematic

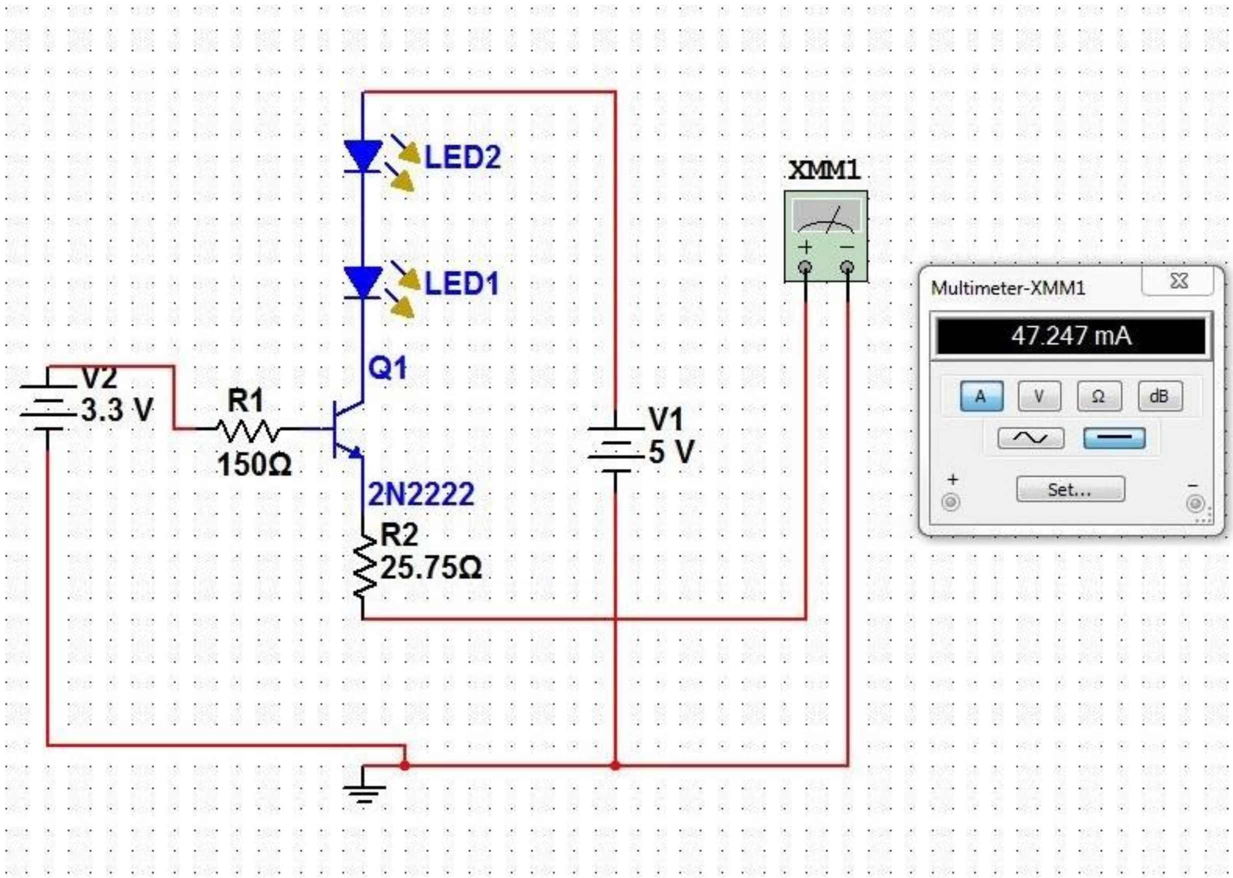


Lighting PCB Layout

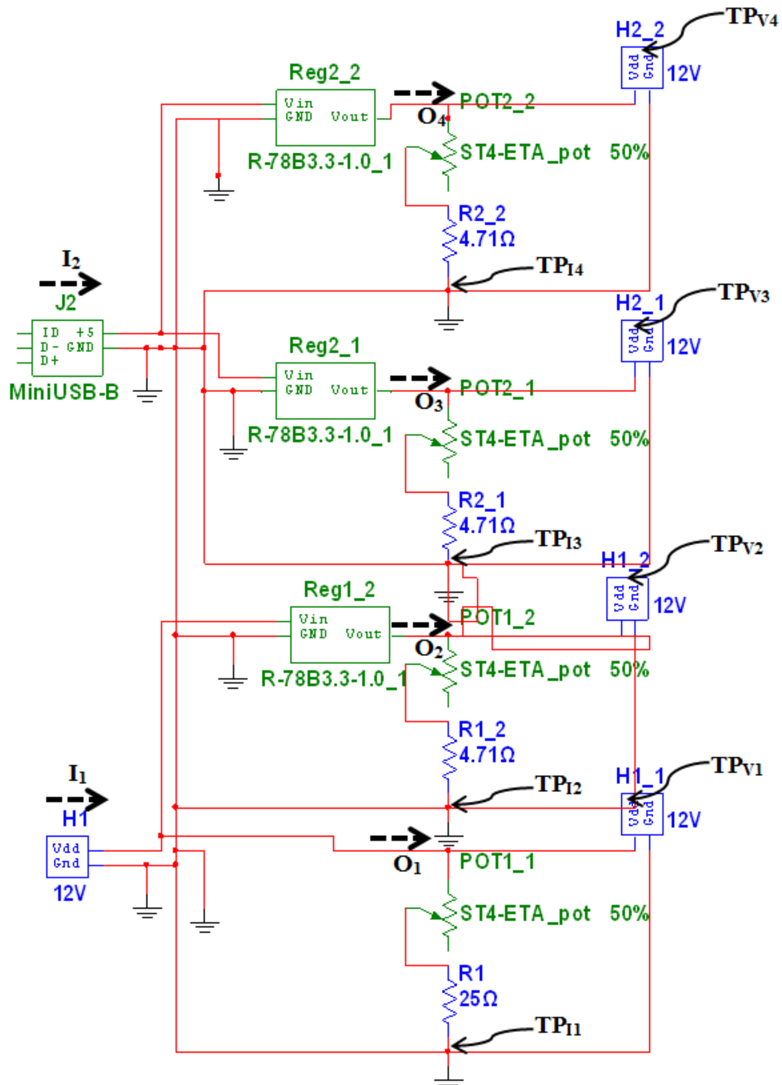


CANBus PCB Layout

–



IR Schematic



Power Supply Schematic

Appendix B (Code) (5 pts)

Click Here for Code!