EE445L - Lab 06 Report

Kevin Gilbert
Gilberto Rodriguez

Professor Bard
Lab: Monday/Wednesday 5-6:15

March 7 2014

Requirements Document

Overview

Objectives

The objectives of this project are to design, build and test a music player. Educationally, students are learning how to interface a DAC, how to design a speaker amplifier, how to store digital music in ROM, and how to perform DAC output in the background. Your goal is to play your favorite song.

Process

The project will be developed using the LM3S1968 board. There will be three switches that the operator will use to control the music player. The system will be built on a solderless breadboard and run on the usual USB power. The system may use the on board switches or off-board switches. A hardware/software interface will be designed that allows software to control the player. There will be at least three hardware/software modules: switch input, DAC output, and the music player. The process will be to design and test each module independently from the other modules. After each module is tested, the system will be built and tested.

Roles and Responsibilities

EE445L students are the engineers and the TA is the client. Students are expected to make minor modifications to this document in order to clarify exactly what they plan to build. Students are allowed to divide responsibilities of the project however they wish, but, at the time of demonstration, both students are expected to understand all aspects of the design.

Interactions with Existing Systems

The system will use the LM3S1968 board, a solderless breadboard, and the speaker. It will be powered using the USB cable. You may use a +5V power from the lab bench, but please do not power the speaker with a voltage above +5V.

Terminology

1. SSI: Synchronous Serial Interface. Communication standard using a shared clock signal, a data signal, and slave select signals.
2. Linearity: Used in reference to the audio amplifier, linearity determines how the amplifier's output matches to its input. If discrete increments of the input lead to discrete increments in the output of an equal rate of growth at a larger value, then the amplifier has a linear function.
3. Frequency Response: Measure of the audio amplifier's range of inputs to output spectrum.

4. Loudness: How loud a tone is. Determined by the amplitude of the output wave, which is in turn determined by the digital value sent to the DAC (larger values provide a higher output voltage).
5. Pitch: Frequency of a wave.
6. Instrument: Type of sound being played in this lab. Can be adjusted by outputing a non-sine wave. Controlling the voltage over time plot of the wave can create new instrumental sounds.
7. Tempo: The speed at which the song is played. Controlled by how often notes are changed.
8. Envelope: The exponential drop in amplitude of the sound waves over time to provide smoother signals.
9. Melody: The primary sequence of notes to form a song's core.
10. Harmony: Accompanying sequence of notes that form a backdrop for a melody.

Security

The system may include software from StellarisWare and from the book. No software written for this project may be transmitted, viewed, or communicated with any other EE445L student past, present, or future (other than the lab partner of course). It is the responsibility of the team to keep its EE445L lab solutions secure.

Function Description

Functionality

If the operator presses the play/pause button the music will play or pause. If the operator presses the play/pause button once the music should pause. Hitting the play/pause again causes music to continue. The play/pause button does not restart from the beginning, rather it continues from the position it was paused. If the rewind button is pressed, the music stops and the next play operation will start from the beginning. There is a mode Lab 5 Music Player and Audio Amp switch that allows the operator to control the volume of the music player. There must be a C data structure to hold the music. There must be a music driver that plays songs. The length of the song should be at least 30 seconds and comprise of at least 8 different sounds. Although you will be playing only one song, the song data itself will be stored in a separate place and be easy to change. The player runs in the background using interrupts. The foreground (main) initializes the player, then executes for(;;){} do nothing loop. If you wish to include OLED output, this output should occur in the foreground. The maximum time to execute one instance of the ISR is 1.798 μs. You will need public functions Rewind, Play and Stop, which perform operations like a cassette tape player. The Play function has an input parameter that defines the song to play. A background thread implemented with output compare will fetch data out of your music structure and send them to the DAC. There must be a C data structure to store the sound waveform, or instrument. You are free to design your own format, as long as it uses a formal data structure (i.e., struct). The generated music must sound beautiful utilizing the SNR of the DAC. Although you only have to implement one instrument, it should be easy to change instruments.

Scope

Phase 1 is the preparation; phase 2 is the demonstration; and phase 3 is the lab report. Details can be found in the lab manual.

Prototypes

A prototype system running on the LM3S1968 board and solderless breadboard will be demonstrated. Progress will be judged by the preparation, demonstration and lab report.

Performance

The system will be judged by three qualitative measures. First, the software modules must be easy to understand and well-organized. Second, the system must employ an abstract data structures to hold the sound and the music. There should be a clear and obvious translation from sheet music to the data structure. Backward jumps in the ISR are not allowed. Waiting for SSI output to complete is an acceptable backwards jump. Third, all software will be judged according to style guidelines. Software must follow the style described in Section 3.3 of the book. There are three quantitative measures. First, the SNR of the DAC output of a sine wave should be measured. Second, the maximum time to run one instance of the ISR will be recorded. Third, you will measure power supply current to run the system. There is no particular need to optimize any of these quantitative measures in this system.

Usability

There will be three switch inputs. The DAC will be interfaced to a 8-ohm speaker.

Safety

If you are using headphones, please verify the sound it not too loud before placing the phones next to your ears.

Deliverables

Reports

A lab report described below is due by the due date listed in the syllabus. This report includes the final requirements document.

Audits

The preparation is due at the beginning of the lab period on the date listed in the syllabus.

Outcomes

There are three deliverables: preparation, demonstration, and report.

Hardware Design

Battery we will use

Li-Ion 18650 Battery: 7.4V 4.4Ah (32.56 Wh, 4A rate, China Cells, 4S/S) Battery Module (2.64)

Your Price: On Promotion: $19.95

In Stock

Product ID #
4894

Part Number: LCH2S2P2WR-China

Lead Time: 5 Business Days!

Quantity: 1 Buy

Add to a new shopping list

Email this page to a friend

***Important Shipping Regulation**	This pack is for testing (prototype) only. It has not been UN38.3 tested yet. Read more...
Packing	• High quality 7.4V Li-Ion rechargeable battery Module is made of 4 pcs High quality Top Brand China 2200mAh cylindrical 18650 cells with PCB • Packed in side by side with 2 series and 2 parallel (2S2P)
Voltage	Voltage: 7.4 V (working) 8.4 V (peak) 5.5 V (cut-off)
Capacity	4400 mAh or 32.56 wh

Protection	• One PCB installed with the battery pack and protects the battery from
	  ◦ Overcharge (>8.4V)
	  ◦ Over-discharge (<5.5V)
	  ◦ Over drain (> 7 Amp)
	  ◦ Short circuits
	• One 4 Amp polyswitch installed to limit max. discharging current and to protect wrong polarity

Pre-wired	6" wire with 18 AWG wire
Charging rate	1 A (recommended), 3 A max.
Max. Discharging Rate	**4 Amp** limited by polyswitch
Dimensions	0.735 x 2.86 x 2.86 inch (Width x Thickness x Length)
Weight	7.0 oz
Smart Tips	• You may choose our Smart 7.4V Li-Ion Battery Charger (1.2A, UL listed) to recharge this battery pack. The estimated charging time is 5.5 hours • If you connect two of these modules in parallel, you can build a 7.4V 8.8Ah Battery pack with a 14 Amp discharge rate.
Applications	• 6V Bike lighting up to 15 W halogen light • External battery pack to run 6V DC devices, such as digital camera
Warning	• Li-Ion Battery may explode if charging or discharging improperly. User must have knowledge on how to charge and discharge Li-Ion battery before making Li-Ion Battery Pack. • For safety warning please see the link here

Battery info

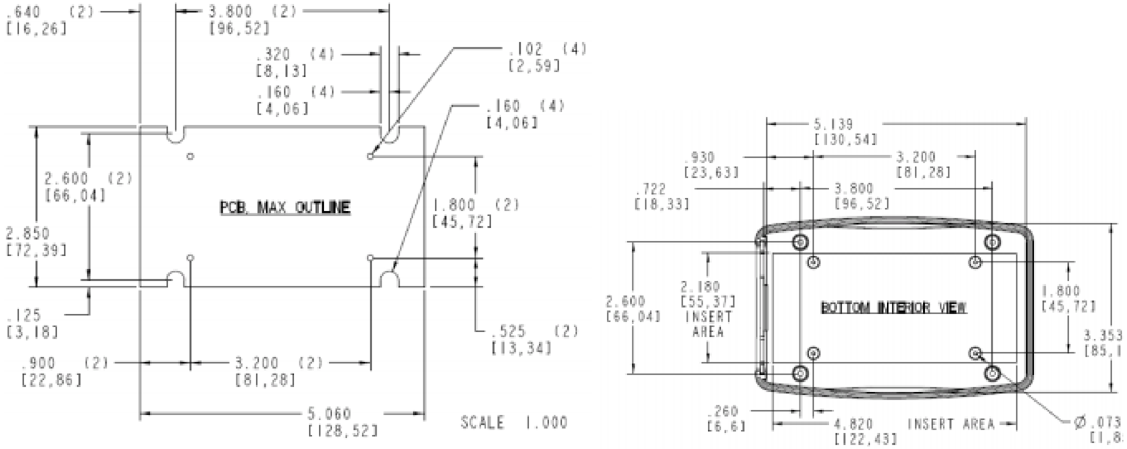


Figure 6.2a. Mechanical drawings for the Pactec XP enclosure.

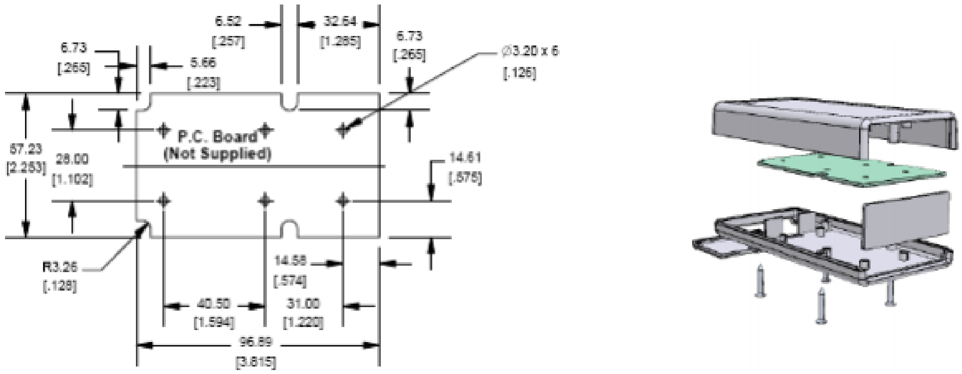
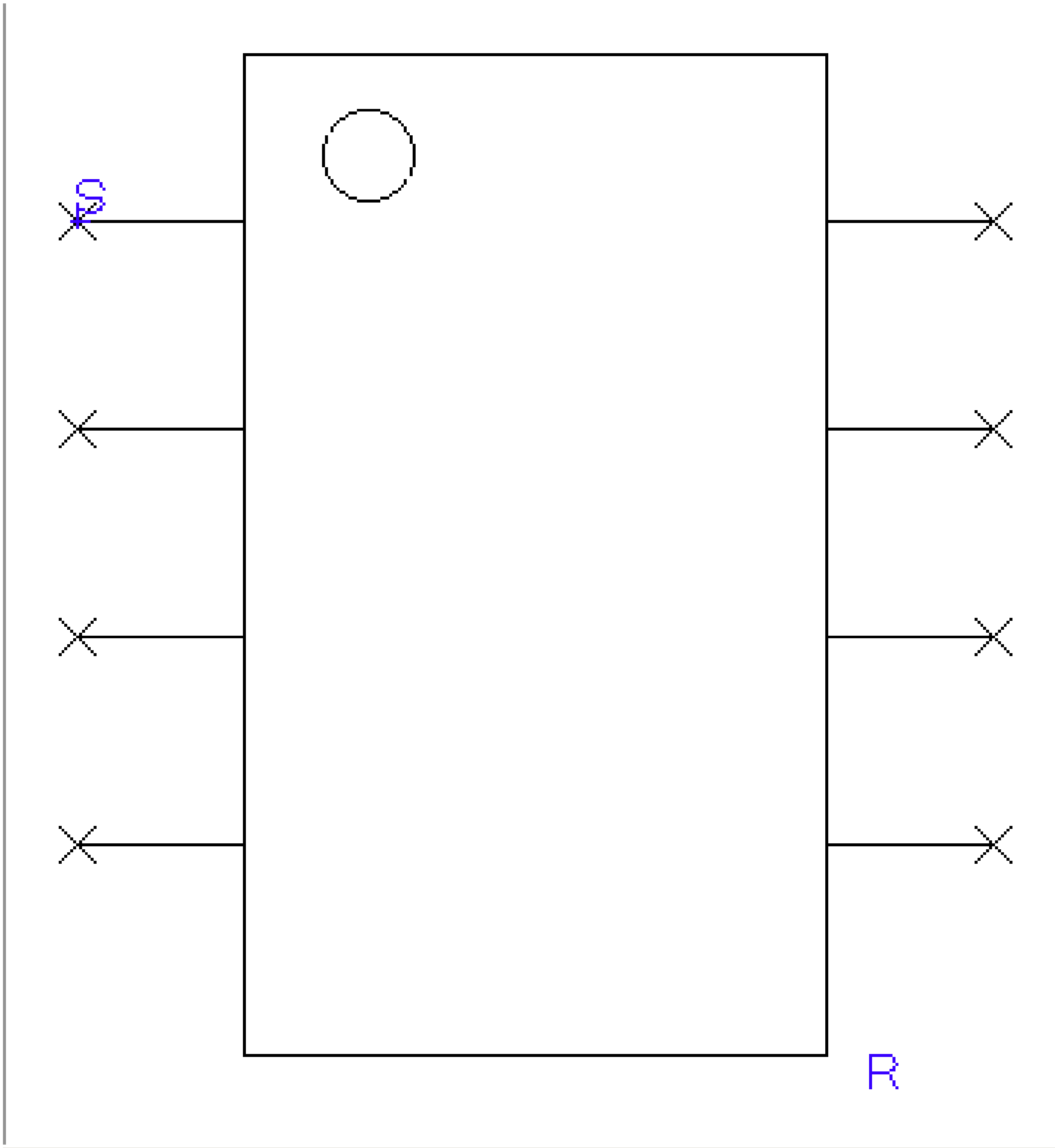
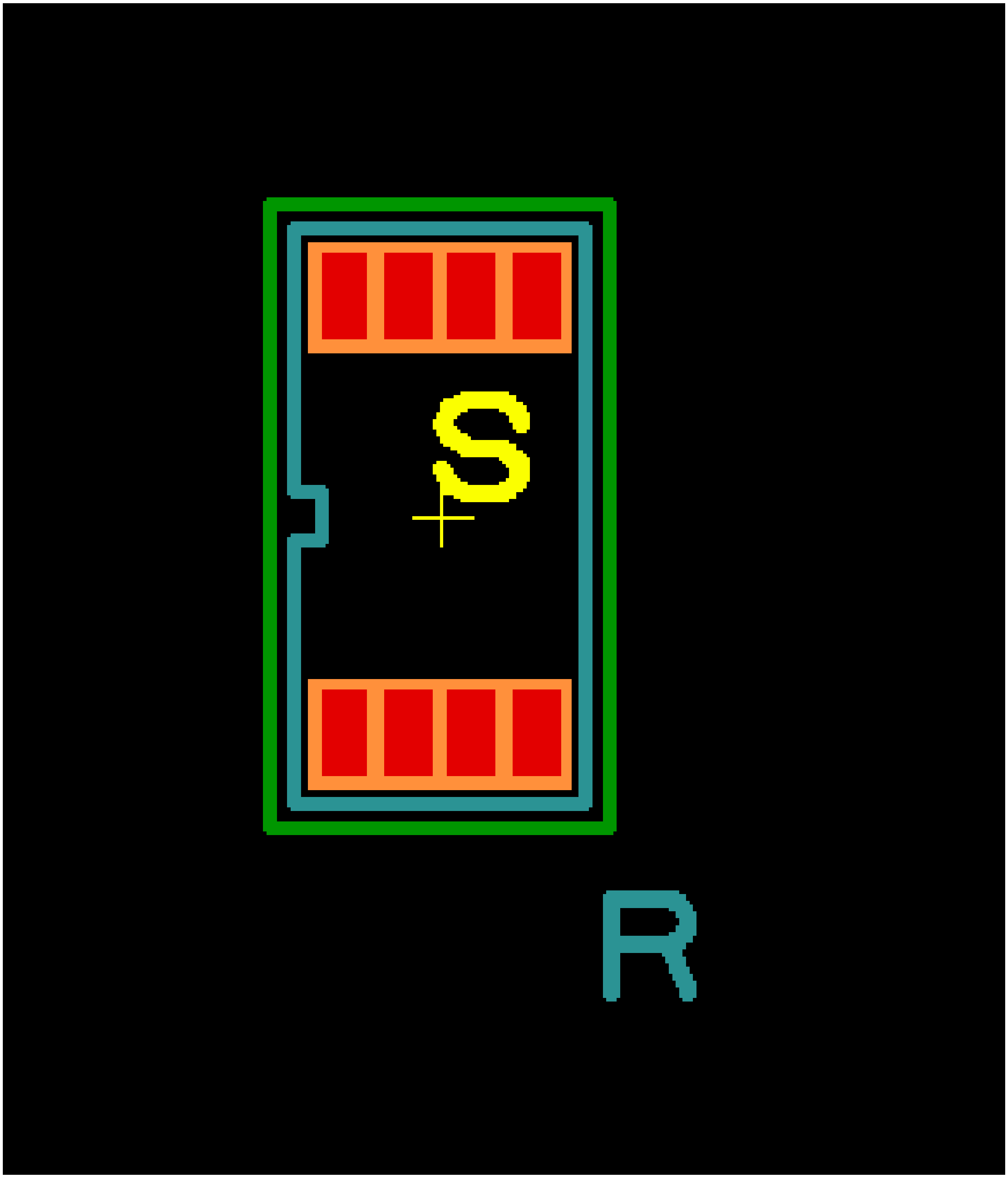


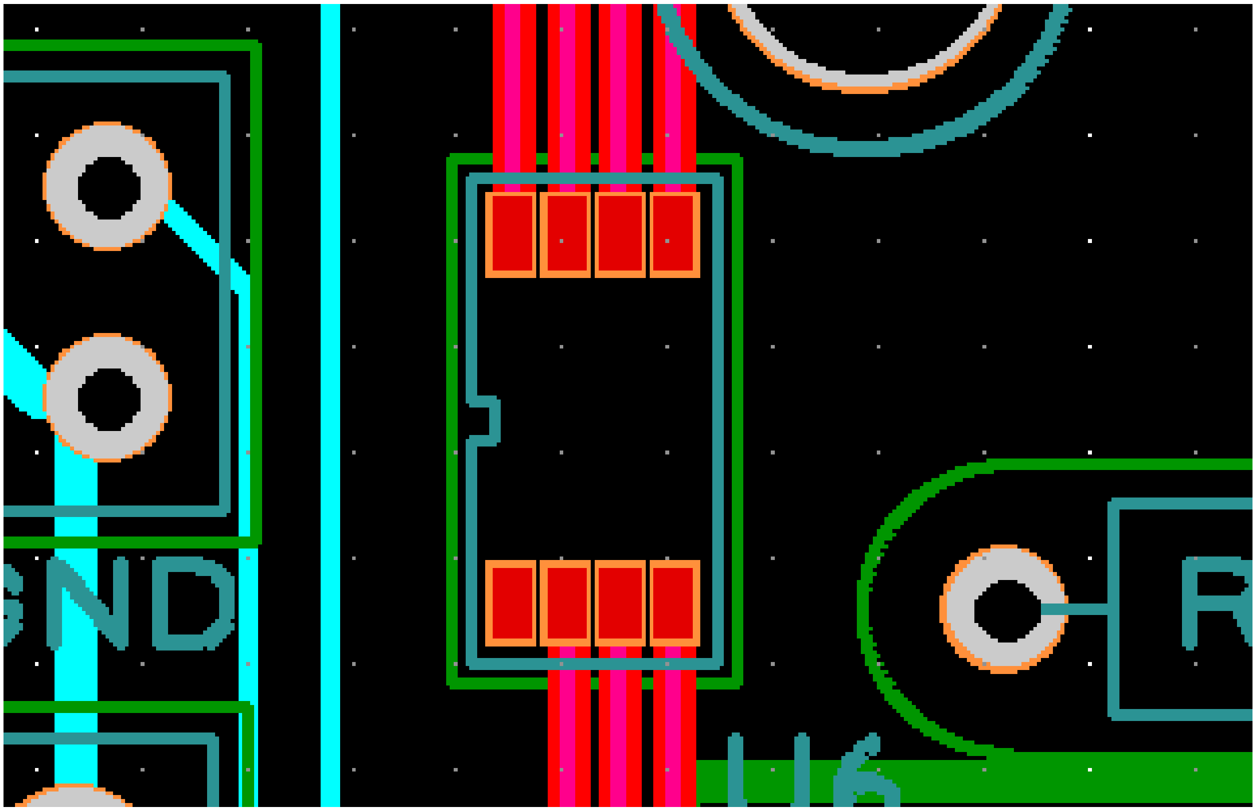
Figure 6.2b. Mechanical drawings for the Hammond 1593Y enclosure.

Used default box

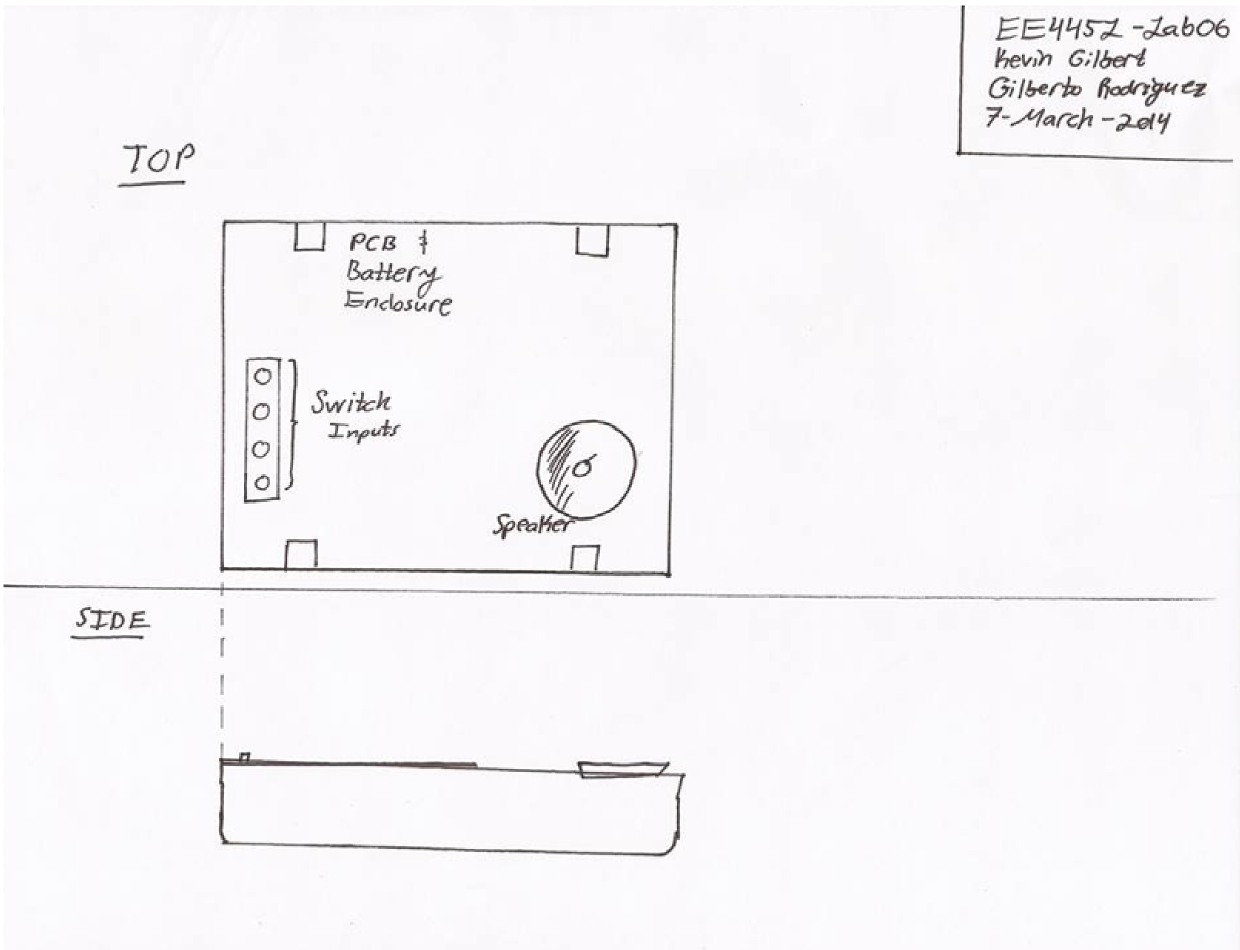


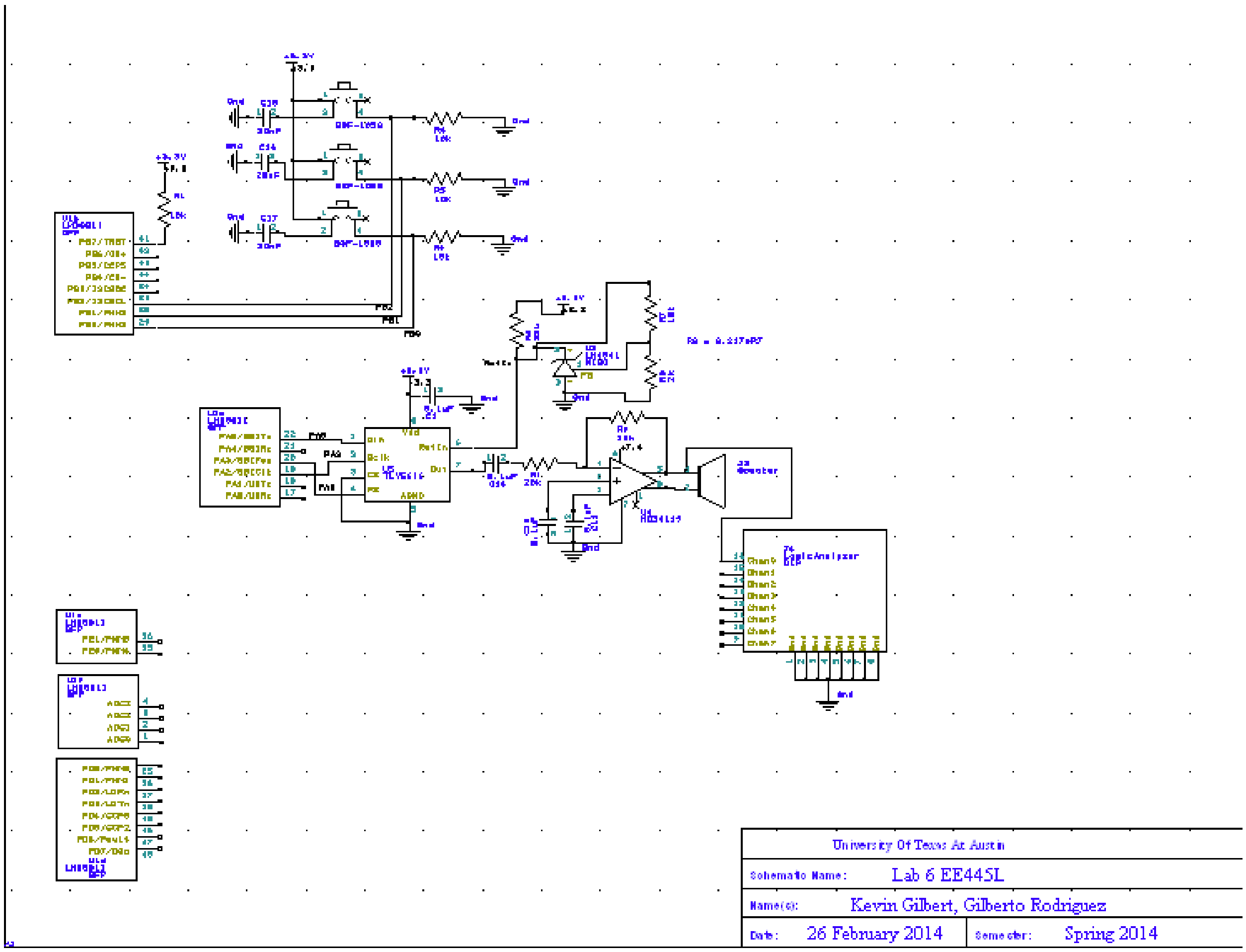
Our custom part: TPA301


S
R

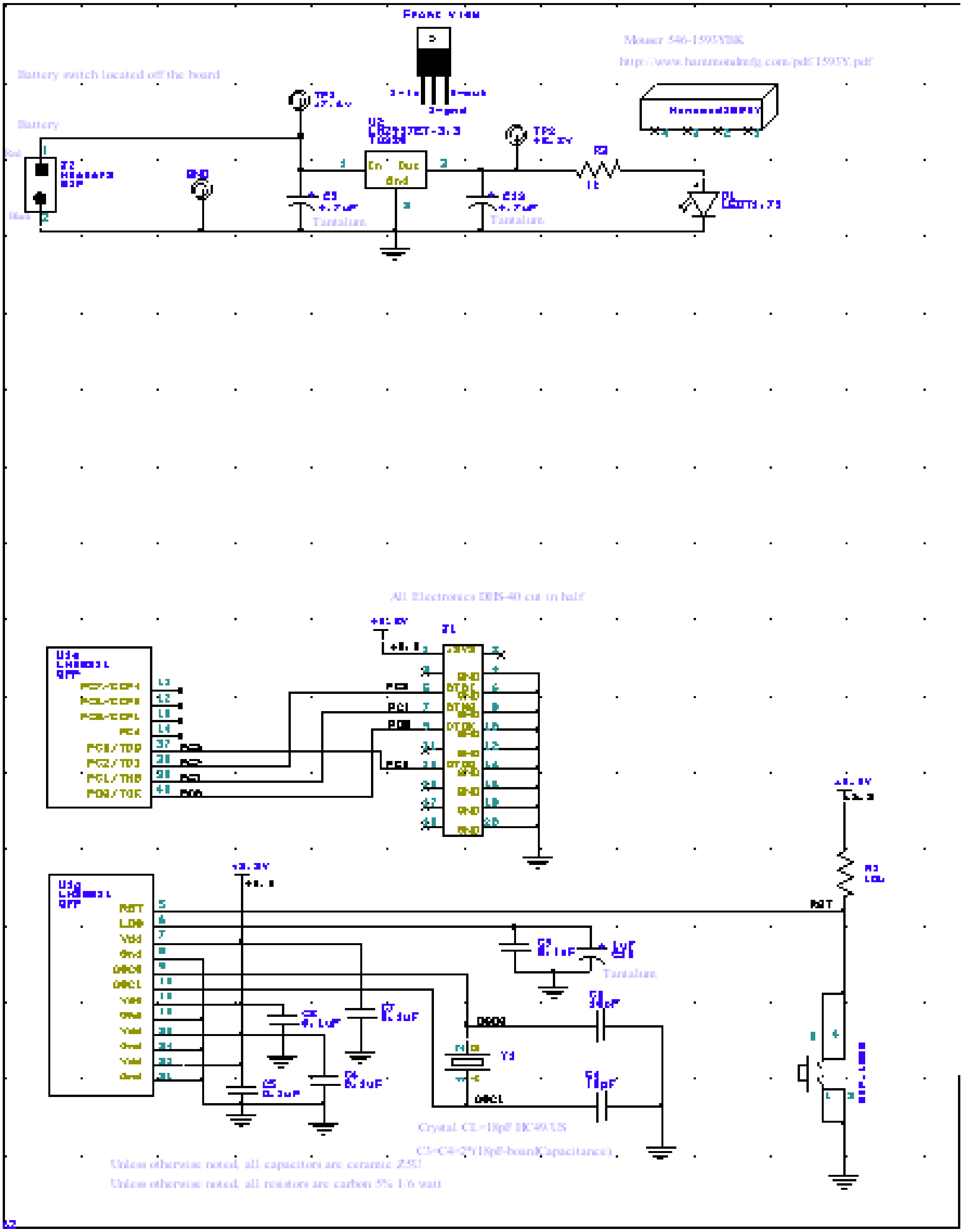


Two mechanical drawings (Procedure 9)

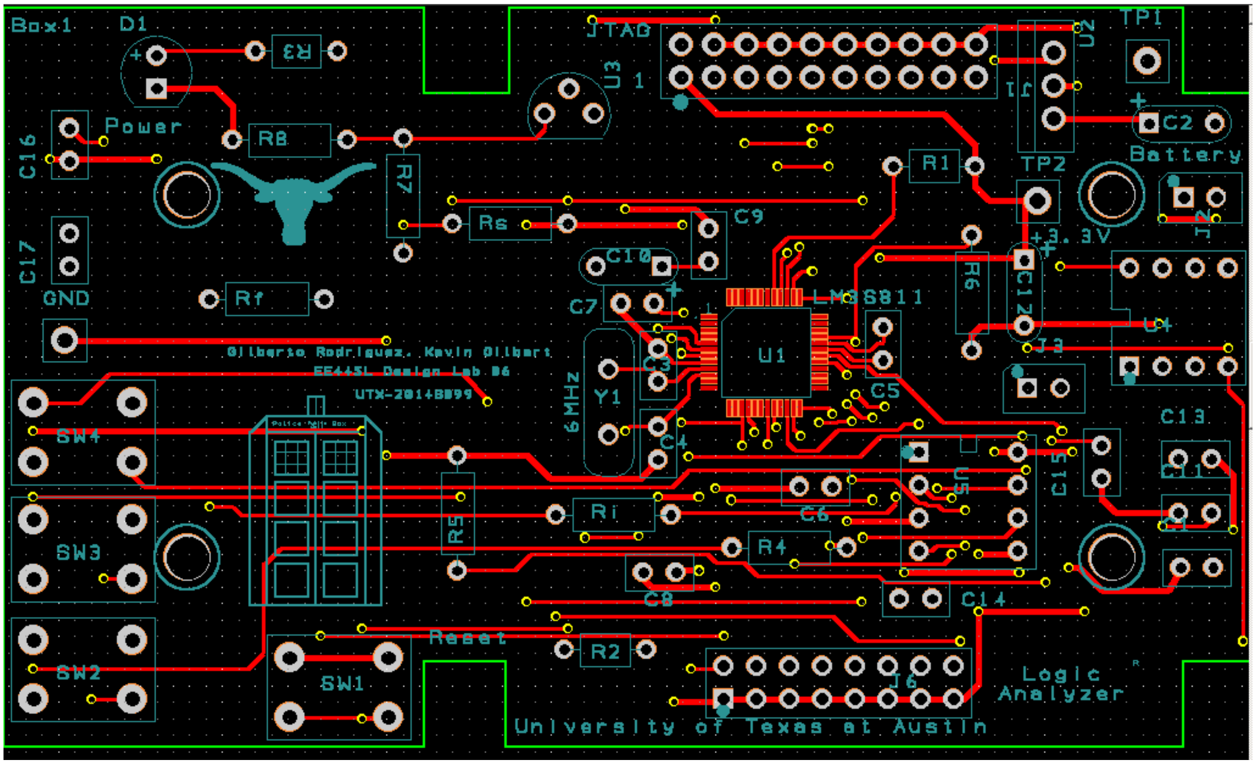

EE4452 -Lab06
Kevin Gilbert
Gilberto Rodriguez
7-March-2014
TOP
PCB &
Battery
Enclosure
Switch
Inputs
Speaker
SIDE

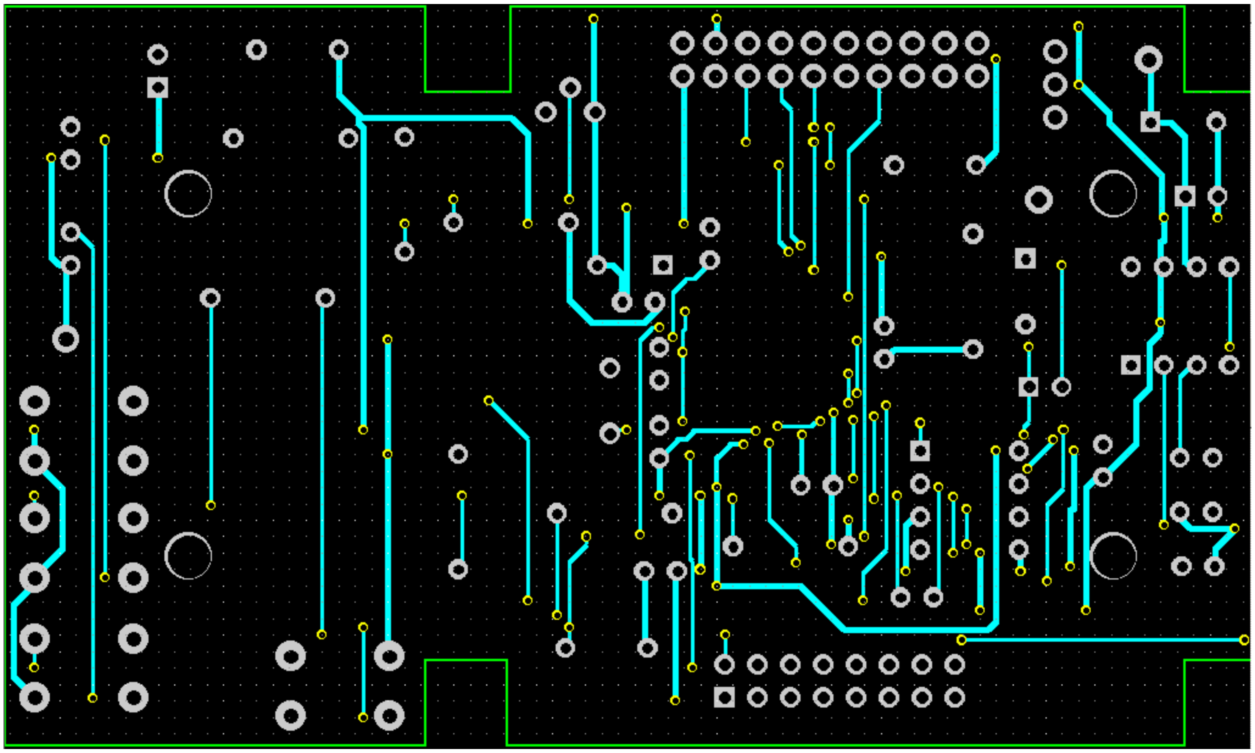


University Of Texas At Austin
Schematic Name:
Lab 6 EE445L
Name(s):
Kevin Gilbert, Gilberto Rodriguez
Date:
26 February 2014
Semester:
Spring 2014



Front View
Mouser 546-1593YBK
http://www.hammondmfg.com/pdf/1593Y.pdf
Battery switch located off the board
Battery
Tantalum
Tantalum
All Electronics DHS-40 cut in half
Tantalum
Crystal: CL=18pF HC49/US
C3=C4=2*(18pF-boardCapacitance)
Unless otherwise noted, all capacitors are ceramic Z5U
Unless otherwise noted, all resistors are carbon 5% 1/6 watt


Box1
D1
R3
JTAG
U3
U2
TP1
Power
R8
J1
C2
C16
R7
R1
TP2
Battery
Rs
C9
+3.3V
J2
C17
C10
R6
C12
GND
Rf
C7
LM3S811
Gilberto Rodriguez, Kevin Gilbert
EE445L Design Lab B6
UTX-2014B899
U1
U4
C3
J3
Y1
6MHz
C5
SW4
C4
C13
C15
U5
C11
Ri
R5
C6
SW3
R4
C8
C14
Reset
R2
SW2
J6
SW1
Logic
Analyzer
University of Texas at Austin



Measurement Data

Build of Materials

Component	Quantity	Price	Vendor	Current	Information
Resistor	5	\$0.08-1	Digi-Key	-	0.25kWatt, 10kΩ
Resistor	2	\$0.08-1	Digi-Key	-	0.12kWatt, 10kΩ
Resistor	4	\$0.08-1	Digi-Key	-	0.25kWatt, 1kΩ(1), 2kΩ(1), 20kΩ(1), 22kΩ(1)
B3F-1050	4	\$0.29-1	Mouser	-	Switchs DIP
Capacitor	4	\$0.35-1	Digi-Key	-	5% NonPolar 0.1μF
Capacitor	3	\$0.35-1	Digi-Key	-	5% NonPolar 20nF
Capacitor	7	\$0.37-1	Digi-Key	-	5% Ceramic 0.1μF(5),10pF(1),18pF(1) [MCU and Clock Filters]
Capacitor	3	\$0.37-1	Digi-Key	-	5% Polar 1μF(1), 4.7μF(2)
Header2	1	\$2.00-1	Samtech	-	SIP Power Pins (J2)
DHS-40 JTAG	1	\$2.00-1	Samtech	-	J1 JTAG Connectors
LED T1.75	1	\$0.52-1	Digi-Key	-	LED (D1)
LM3S811 QFP	1	\$6.39-1	Newark	-	MCU
LM2937ET-3.3	1	\$0.92-1 - 1kU	TI	5mA	TO220 (Voltage Regulator U2)
LM4041	1	\$0.29-1	Digi-Key	-	Shunt Voltage Reference (U3)
Pins	1	\$2.00-1	Samtech	-	Logic Analyzer Headers DIP (J6)
Hammond-1593Y	1	\$4.82-1	Mouser	-	Box (Box1)
MC34119	1	\$0.50-1	Digi-Key	-	Audio Amplifier DIP (U4)
TLV5616	1	\$4.08-1	Digi-Key	-	DAC DIP (U5)
Clock-XTAL	1	\$1.60	Digi-Key	-	DSC Y1
Battery	1	\$19.85-1	Batter Space	-	7.4V Li-Ion Battery

Bill of Materials (quantity, package type, cost, and supply current) (Procedure 2)

Our music player uses a current of 145mA, so we chose a battery that could power the music play for 24 hours while it is running. The battery we chose has a capacity of 4400mAh, which can power our music player for approximately 30 hours.

Analysis and Discussion

Using the onboard JTAG pins, we would flash code to test the capabilities of each device through a set of testing points. To make the process stream-lined, these testing points would have traces connected to a set of logic analyzer pins. These pins can be monitered during an automated testing procedure to verify the operation of the device. Specically, speaker output is measured in our current design; however, additional test points at the DAC output, DAC inputs, button inputs, and audio amplifier connections could be tested as well. Specifically, a 8-pin digital analyzer attached to these points would allow for observing the response of the system. The software component would generate several sine waves of varying frequencies