Ronald Macmaster and Parth Adhia

EID: Rpm953 and Pda375

10/14/2016

EE 445L – Lab 6: Introduction to PCB Design

1.0: Objectives:

We will explore the aspects of PCB layout by designing a PCB layout for our Lab 5 Music Player. Consequently, it will be important for us to understand the systems-level approach to designing an embedded system. That will require us to consider various mechanical aspects of our system, the availability of electronic parts, the cost of our embedded system, and the power requirements of our embedded system. Also, we will undergo organized forward planning so we can design a system that is not only functional, but also testable throughout the full design cycle.

Lab5 Revised Requirements document

1. Overview

1.1. Objectives: Why are we doing this project? What is the purpose?

Our primary objective for this project is to design, build and test a music player. We are learning how to interface a DAC, design and test a speaker amplifier, store digital music in ROM, and perform DAC output in the background.

Additionally, we will develop two or three low-level device drivers for our TLV5616 DAC to communicate with it via SSI protocol. We will also design a hardware / software interface for a couple of switch inputs. We will measure the supply current necessary to operate our embedded system, and implement our music player with periodic timer interrupts.

Our project will also require us to design data structures to represent music loops and instrument sounds. Finally, we will explore the deeper design principles of signal communication with spectral measurements and observations.

1.2. Process: How will the project be developed?

The project will be developed using the TM4C123 board. There will be two or 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.

1.3. Roles and Responsibilities: Who will do what? Who are the clients?

Ronald Macmaster and Parth are the engineers and the Dylan Zika (TA) is the client. We have modified this document to clarify exactly what we plan to build. We are allowed to divide responsibilities of the project however we wish, but, at the time of demonstration, both of us are expected to understand all aspects of the design.

Who will do what?

Ronald Macmaster will be responsible for drafting the requirements document and the lab report. He will also create the software design diagrams that will consist of a call graph and a data-flow graph. He will draft up the interface files for the various software modules and create program skeletons for the various driver files. Finally, he will supervise and contribute to the software development process and submit all of the assignment documentation.

Parth Adhia will be responsible for drafting the hardware design documentation. This will require a schematic diagram of the external system hardware that should be built using the PCB Artist program. He will write a significant portion of the software drivers and assemble the external system hardware circuit. He is responsible for providing the speaker and tactile switches as well. Finally, he will edit and contribute to the various project documentation as fit.

Both engineers will contribute equally to the schematic layout. The schematic layout will require both engineers to research the datasheets for the TLV5616, TPA731, LM4041, and the TM4C123.

1.4. Interactions with Existing Systems: How will it fit in?

The system will use the TM4C123 board, a solderless breadboard, and the speaker as shown in Figure 5.1. It will be powered using the USB cable. **We will use a +5V power from the lab bench, and we will not power the TPA731 or the speaker with a voltage above +5V.**

1.5. Terminology: *We herein define the following key terms*:

SSI (Synchronous Serial Interface):

The **SSI** system allows microcontrollers to communicate synchronously with peripherial devices or other controllers over a serial protocol. In an SPI implementation, two devices operate over a communicated and synchronized clock signal.

Linearity:

Linearity a mathematical relationship that implies a signal can be graphically represented as a straight line. One quantitative measure of linearity is the linear regression correlation coefficient. Given a function f(n), the output is linear if $f(n+1) – f(n) = f(m+1) – f(m) = dx$ for all n and n.

Frequency Response:

The **Frequency Response** is a quantitative measure of the system's output signal frequency spectrum in response an input signal. It is often used to describe the dynamic behavior of an LTI system.

Loudness:

The **loudness** of the sound wave is our psychological perception of the wave's physical strength. It is correlated to the magnitude of the sound wave's amplitude.

Pitch:

Pitch is another property of sound that allows the perception of ordering on a frequency-based scale. Pitch provides our means to perceive notes as "higher" or "lower" in comparison to other notes.

Instrument:

An object that is created or adapted in order to create musical sounds. Anything that produces sound can be considered a musical instrument, but we usually reserve this classification for things like a flute, piano, guitar, or violin.

Tempo:

In the world of music, the **tempo** of a song defines the speed of a song. It is usually calculated in beats per minute (bpm).

Envelope:

The **envelope** of a signal is the smooth curve that outlines its amplitude extremes. It can be considered the relationship of amplitude structure to time.

Melody and Harmony:

The **melody** of a song is the combination of rhythm and pitch that gives it its primary identity. You recognize a song by the tune of its melody. The **harmony** of the song is the lower, supporting role to the melody that adds completeness to the song. The melody alone sounds empty without the harmony.

1.6. Security: How will intellectual property be managed?

The system may include software from StellarisWare and from the book. No software written for this project will 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 our team to keep its EE445L lab solutions secure. **We will manage our source code over a private git repository hosted by GitHub Inc.**

2. Function Description

2.1. Functionality: What will the system do precisely?

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 switch that allows the operator to control some aspect of the player. **Pressing the mode switch will toggle the music player's playlist feature (switch the song)**. **The tempo of the song can also change through the switch interface.**

There will 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 frequencies. **Although we 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 **while(1){}** do nothing loop. Any optional LCD output should occur in the foreground. The maximum time to execute one instance of the ISR is **2.8 microseconds**. We 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. We 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 we only have to implement one instrument, it should be easy to change instruments.

2.2. Scope: List the phases and what will be delivered in each phase.

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.

For preparation, we will deliver an updated requirements document, schematic diagram of our hardware system, and the call and data-flow graph that models our software design. We will also demonstrate possession of the necessary hardware components and reasonable progress on the various software modules. The software will be written and compiled by the preparation date.

For checkout demonstration, we will begin by demonstrating the features of our DAC Driver. We will also present our completed music player software and system. The music player will be stand-alone (powers on and off) and the speaker will be placed in a box.

The final documentation for this project will be written up in a lab report and submitted over Canvas for grading by Midnight on Friday, October 7th.

2.3. Prototypes: How will intermediate progress be demonstrated?

A prototype system running on the TM4C123 board and solderless breadboard will be demonstrated. Progress will be judged by the preparation, demonstration and lab report.

2.4. Performance: Define the measures and describe how they will be determined.

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 Kerrigan and Ritchie's book, *The C Programming Language.* 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, we will measure power supply current to run the system. There is no particular need to optimize any of these quantitative measures in this system.

2.5. Usability: Describe the interfaces. Be quantitative if possible.

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

The TM4C123 and TLV5616 will interface through a custom SPI protocol. The SSI Clock will be set to 10Mhz during operation. The output of the TLV5616 will flow into the TPA731, and the TPA731 will amplify the output to our speaker.

2.6. Safety: Explain any safety requirements and how they will be measured.

If you are using headphones, please verify the sound it not too loud before placing the phones next to your ears. Connecting or disconnecting wires on the protoboard while power is applied may damage the board.

3. Deliverables

3.1. Reports: How will the system be described?

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

The final lab report is due on Friday, October 7th at Midnight. This report will include the final requirements document. The outline of the lab report is as follows:

A) Objectives (final requirements document)

B) Hardware Design (External hardware schematic to interface TLV5616)

C) Software Design (Updated software modules, Call graph, and Data-flow graph)

D) Measurement Data (Data and calculated resolution, range, precision, and accuracy of the DAC, experimental response of the DAC, debugging profile, +5V voltage and RMS voltage, and current required with and without music playing.)

E) Analysis and Discussion (short answers to a couple design questions)

3.2. Audits: How will the clients evaluate progress?

Clients will be presented with a preparation the week before before the final checkout. The lab report will be presented at conclusion of the project.

3.3. Outcomes: What are the deliverables? How do we know when it is done?

There are three deliverables: preparation, demonstration, and report. **Most of our work and design will be documented in the final lab report.**

2.0 Hardware Design:

Battery Selection

In order to power our Lab5 music player, we will need either a 3.7V Li-ion battery or 4.7 NiMH battery. The system will require a 3.3V regulator for our interface between the processor and DAC. We will power the TPA731 from 4.8 / 7.4V.

We have decided to go with two ***Tenergy 3.7V 2600mAh Lithium-Ion 18650 Flat Top Rechargeable Batteries (MH48285).*** They can be added in series to provide the 7.4V power supply for our TPA731 amplifier. Our lab 5 circuit requires 131mA of current to run in the worst case. These batteries provide a 2600mAh energy supply to our circuit which implies a 20 hour lifetime in the worst case. However, this current measurement includes the power to run the LCD. If we absolutely must meet a hard 24 hour lifetime requirement, we may remove the LCD from our music player.

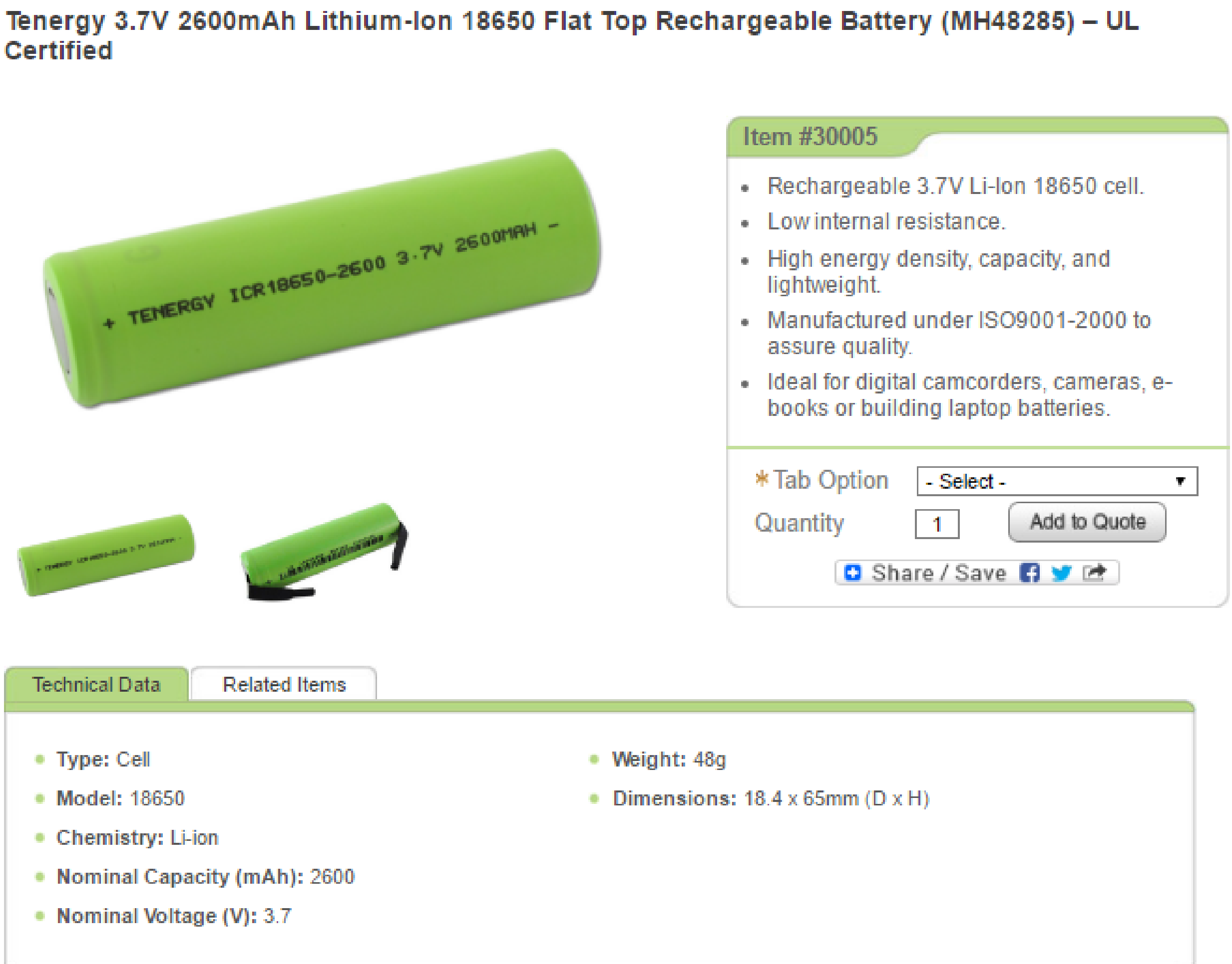
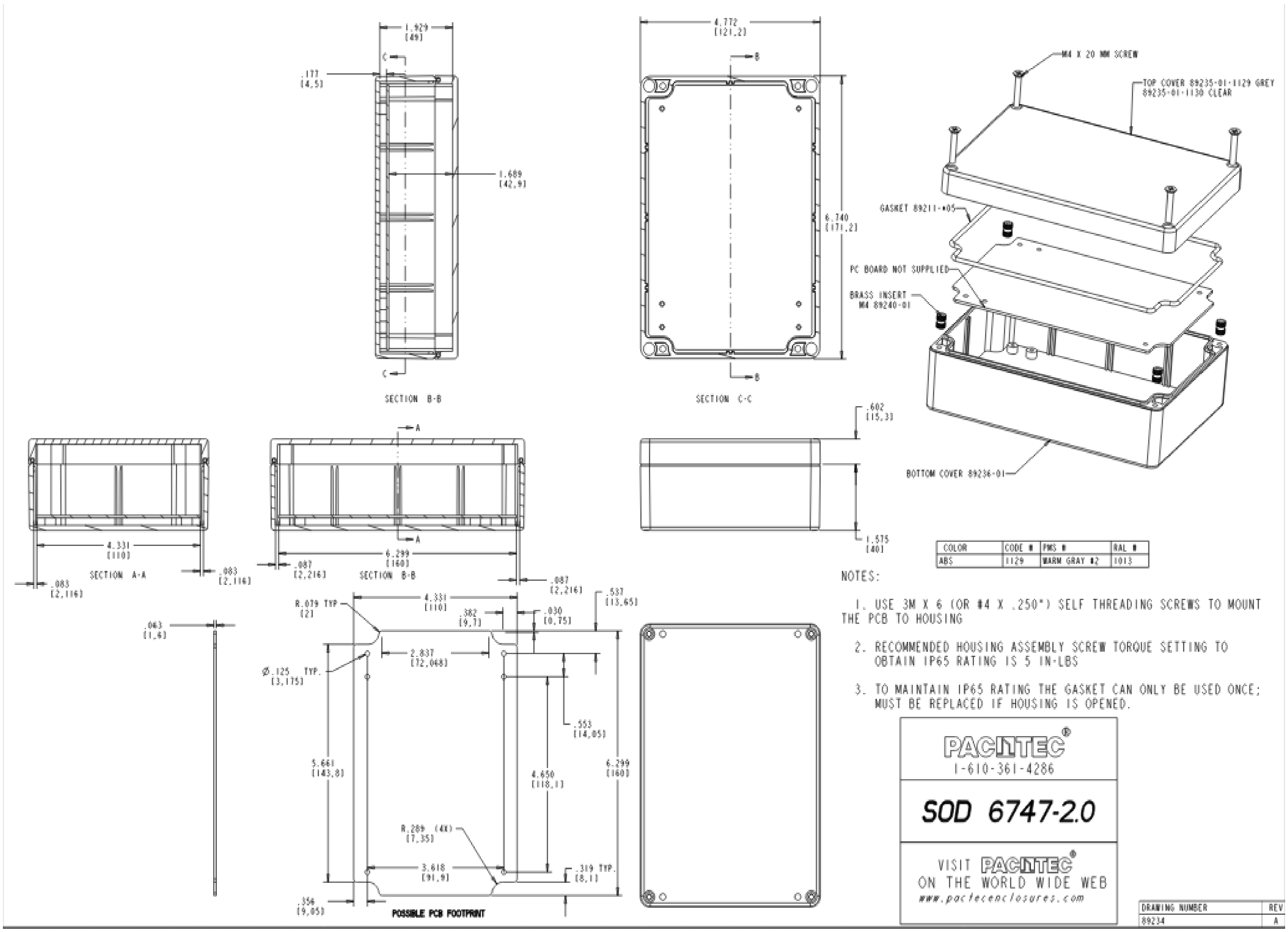


Figure 2.1: *Battery specifications for our Tenergy 3.7V Li-ion battery. The capacity is 2600mAh, voltage is 3.7V, weight is 48g, and dimensions are 18.4 x 65mm. Our music player requires two of these batteries. Each battery is $7.40 on Amazon with a $6.01 shipping fee for a purchase of two.*

Enclosure Selection

Our approximate PCB Dimensions are 4.10 inches by 2.45 inches. However, we will also want to enclose the battery and the speaker with our PCB circuit. Each battery is about 0.23 inches by 0.70 inches, so we will need a box that is bigger than 4.50 inches by 2.50 inches. Along with the PCB, speaker, and batteries, we will need to attach the ST7735R LCD to the front of the box. Buttons and a headphone jack may also protrude the edges of the box.

We have decided to go with the SOD6747-2.0 Kit box from PACTEC Enclosures. The PCB box is 6.74 x 4.77 x 2.18 in., and it can support a PCB with dimensions up to 5.66 x 4.33 in. This will be more than enough room to encapsulate the PCB, batteries, and speaker into our final enclosure. The box will cost a total of $14.48. Fitted PCB drawings for the enclosure are presented below.



COLOR	CODE #	PMS #	RAL #
ABS	1129	WARM GRAY #2	1013

Figure 2.2: *Drawing specifications for our SOD6747-2.0 enclosure kit. The box's external dimensions are 6.74 x 4.77 x 2.18 in., and the PCB dimensions are 5.66 x 4.33 in. We will enclose our speaker and Li-ion batteries along with our PCB.*

SCH/PCB Component Design

We created a PCB artist component for the HCF4056M013TR.
The HCF4056 is a BCD to 7-Segment Display Decoder that acts as a driver between a BCD input and 7-Segment output. The decoder converts a binary coded decimal input to a 7-segment display output. It allows the input signal voltages to differ from the 7-segment output voltages, and it can also function as a voltage driver for the display. The decoder can be configured to use either positive logic or negative logic.

Demonstrated below are the Schematic and PCB symbols we created for the component.

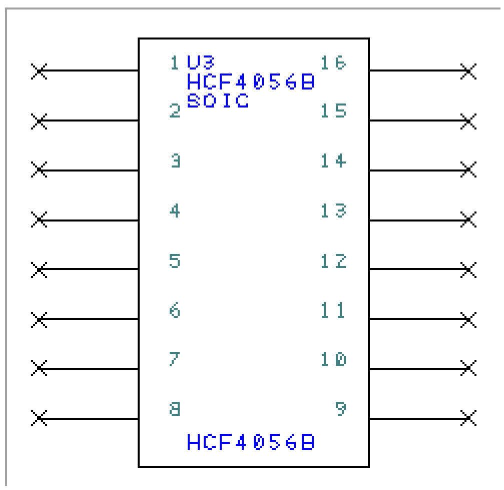


Figure 2.3: *SCH Symbol created for our HCF4056M013TR. The symbol is included in the decoder.ssl library*

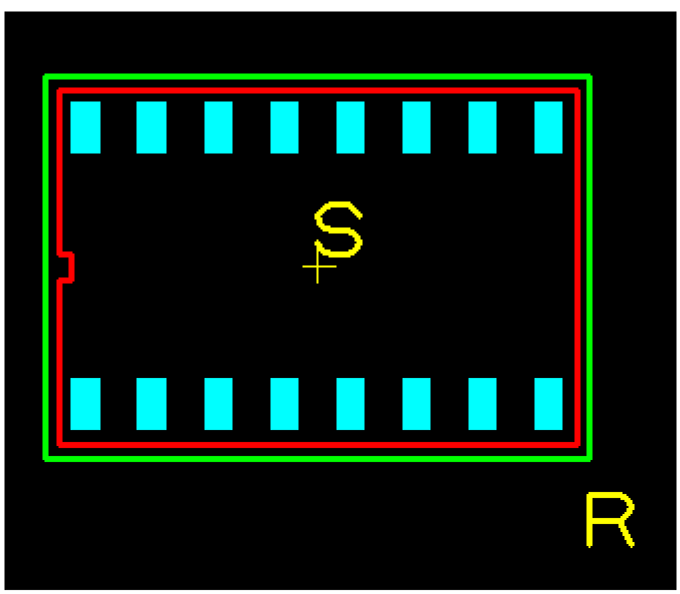


Figure 2.4: *PCB Symbol created for our HCF4056M013TR. The symbol is included in the decoder.pcl library*

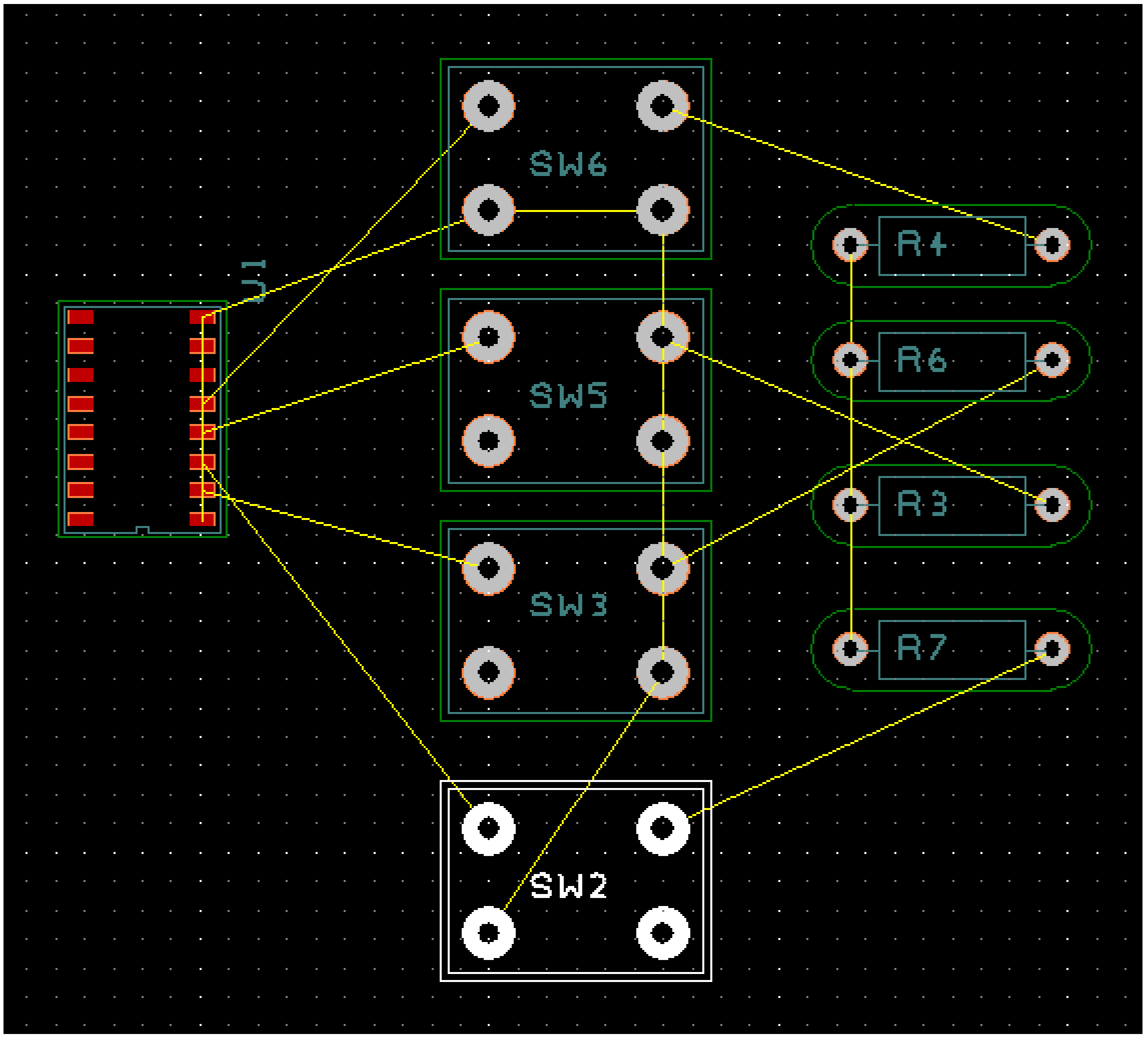


Figure 2.5: *Example PCB Utilizing our HCF4056M013TR component. The left outputs can be wired up to a 7-Segment Display.*

Two Mechanical System Drawings

The LCD of our music player interface perturbs out of the front of the box. The top layer of the enclosure is secured with four corner screws. Inside of the box, the PCB is secured to the left with six custom-drilled holes. The 3.7V battery pack separates the PCB board from the speaker on the far right of the enclosure. There is a strong and secure bottom layer of the enclosure. That holds everything together.

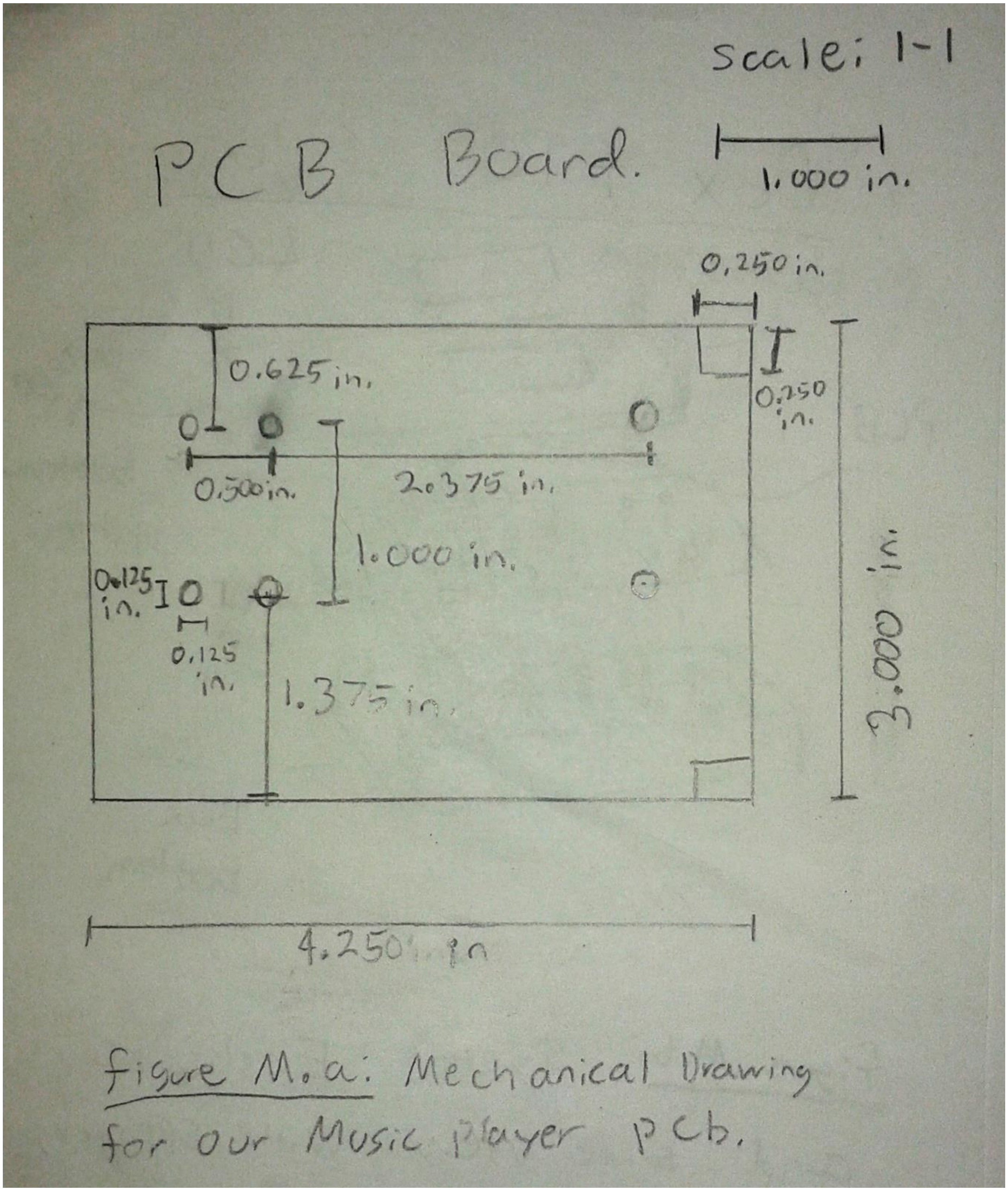


Figure 2.6a: *Mechanical drawing detailing the PCB mechanical measurements. The figure is drawn with a 1-1 scale. Our PCB measures 4.250 x 3.000 inches.*

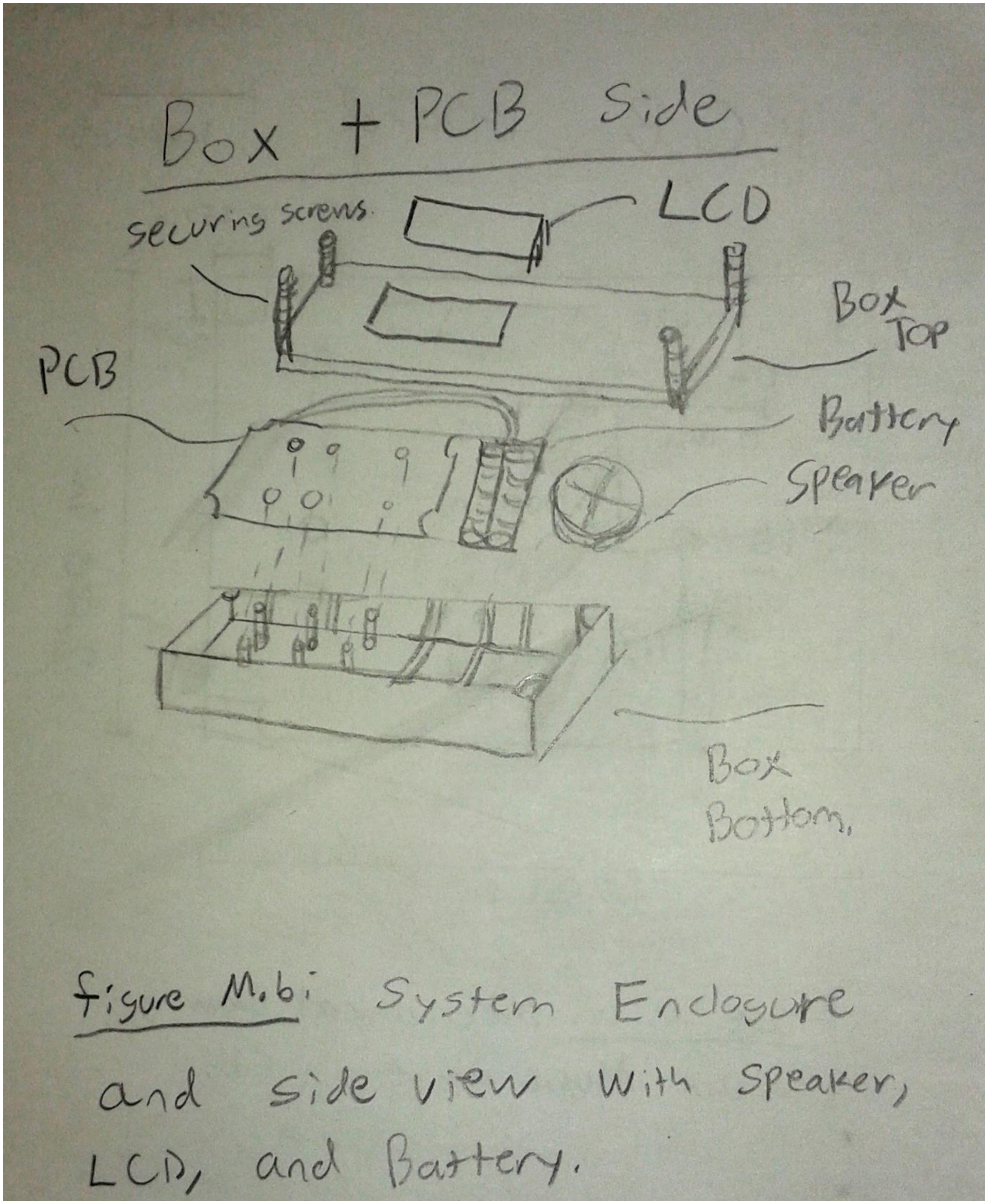


Figure 2.6b: *Mechanical drawing detailing the entire system and enclosure from a side view. The LCD is embedded at the top of the box, and the PCB, battery pack, and speakers are stored on the inside.*

Final circuit diagrams of the embedded system, (SCH files)

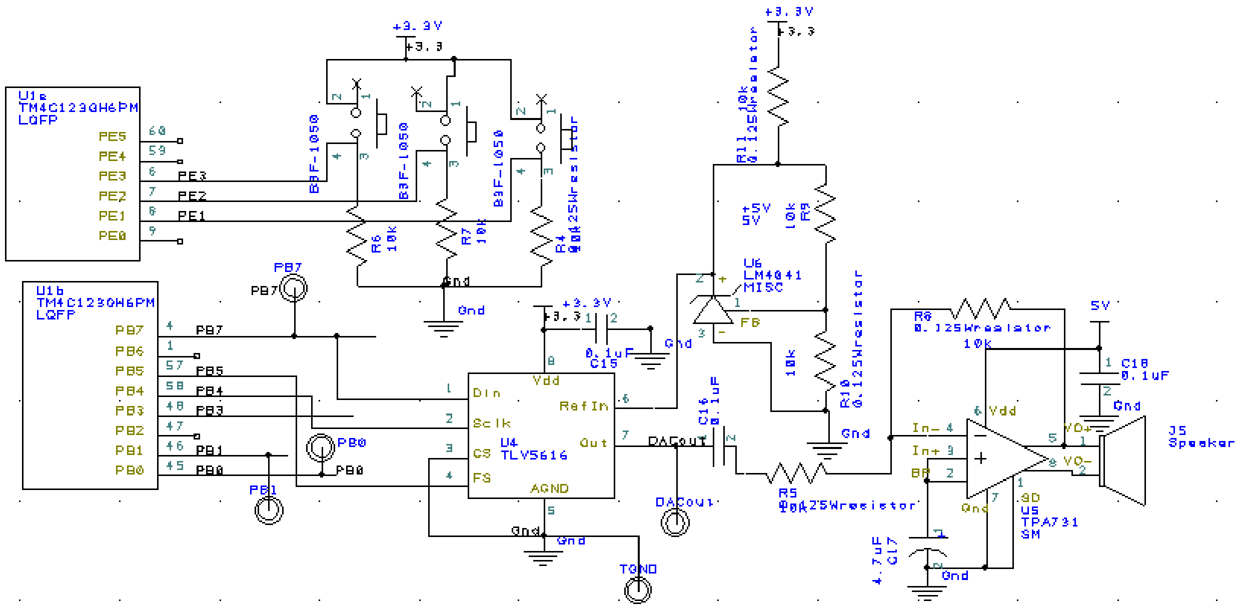


Figure 2.7: *Hardware schematic for our music player system. The TLV5616 is the DAC interfaced through SSI. The DAC output is directed into the TPA731 amplifier which is then delivered to the 8-Ohm speaker.*

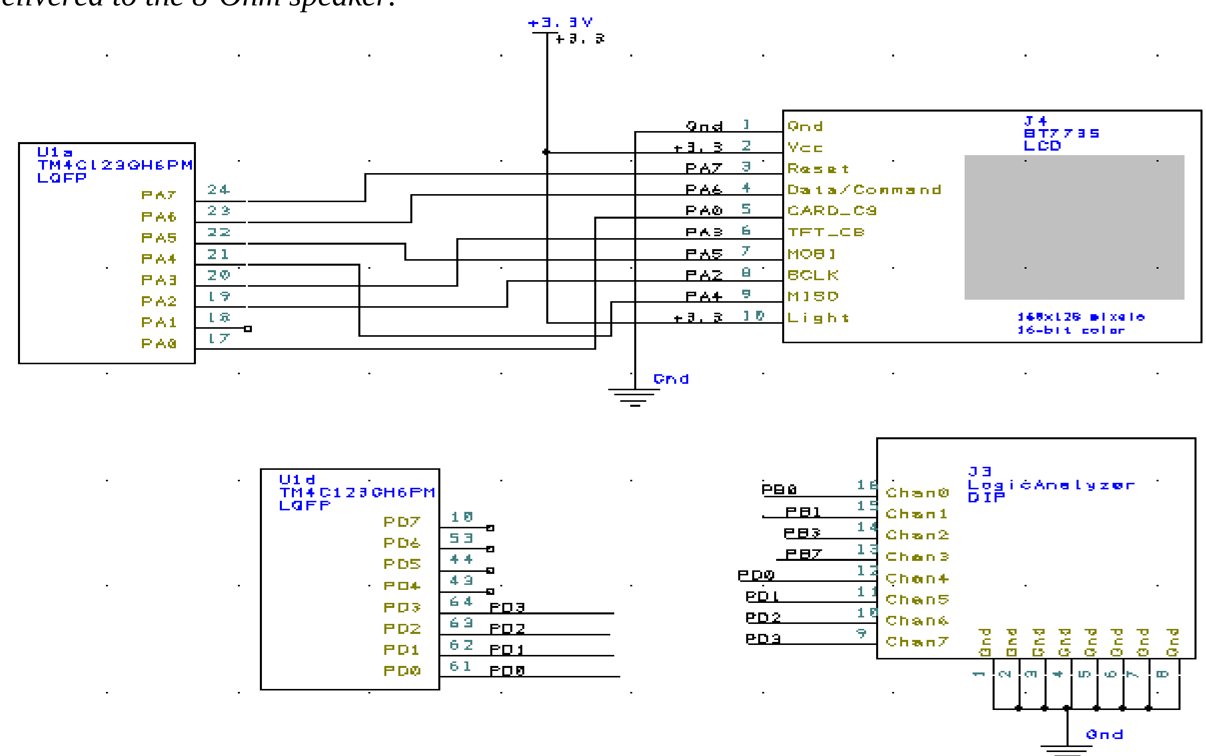


Figure 2.8: *Hardware schematic detailing the LCD and Logic Analyzer setups on Port A and Port D. The logic analyzer also shares pins with Port B.*

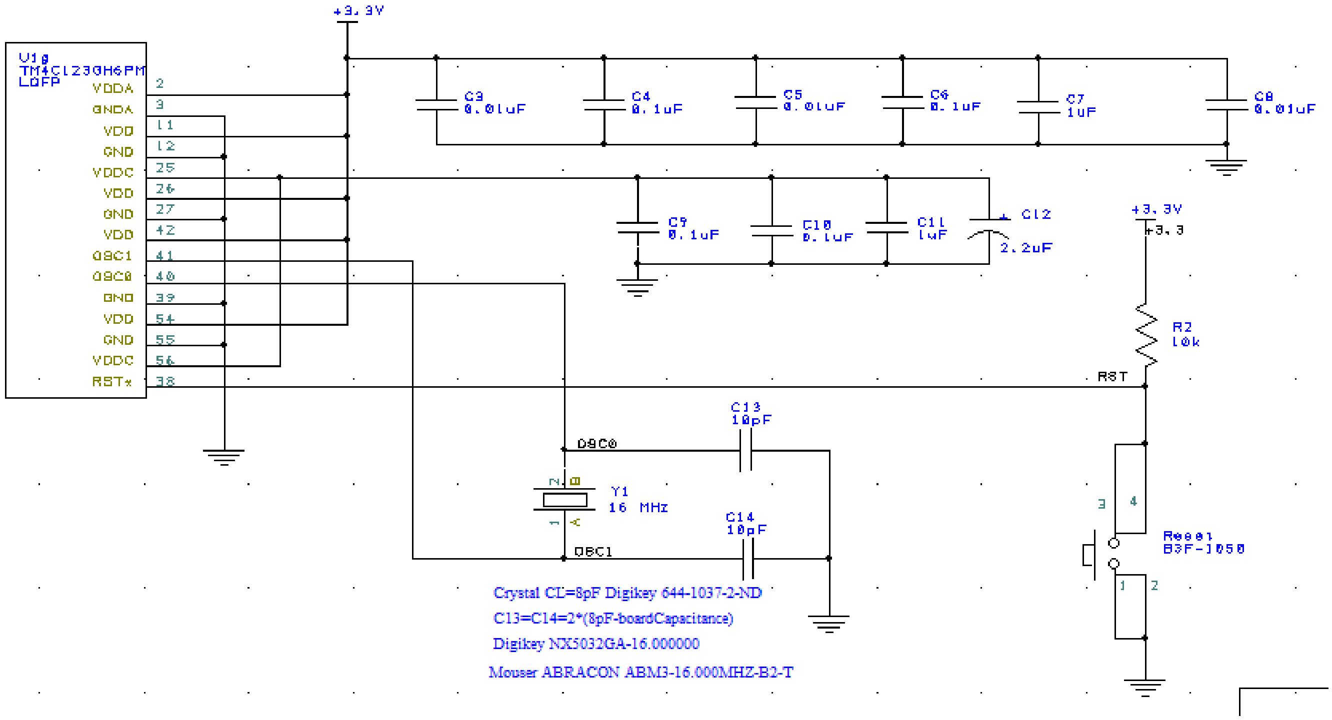


Figure 2.9: *Layout for the Power Supply capacitors, Reset Switch, and Crystal Oscillator. C13 and C14 (10pF capacitors for crystal) are very important capacitors with low tolerance values.*

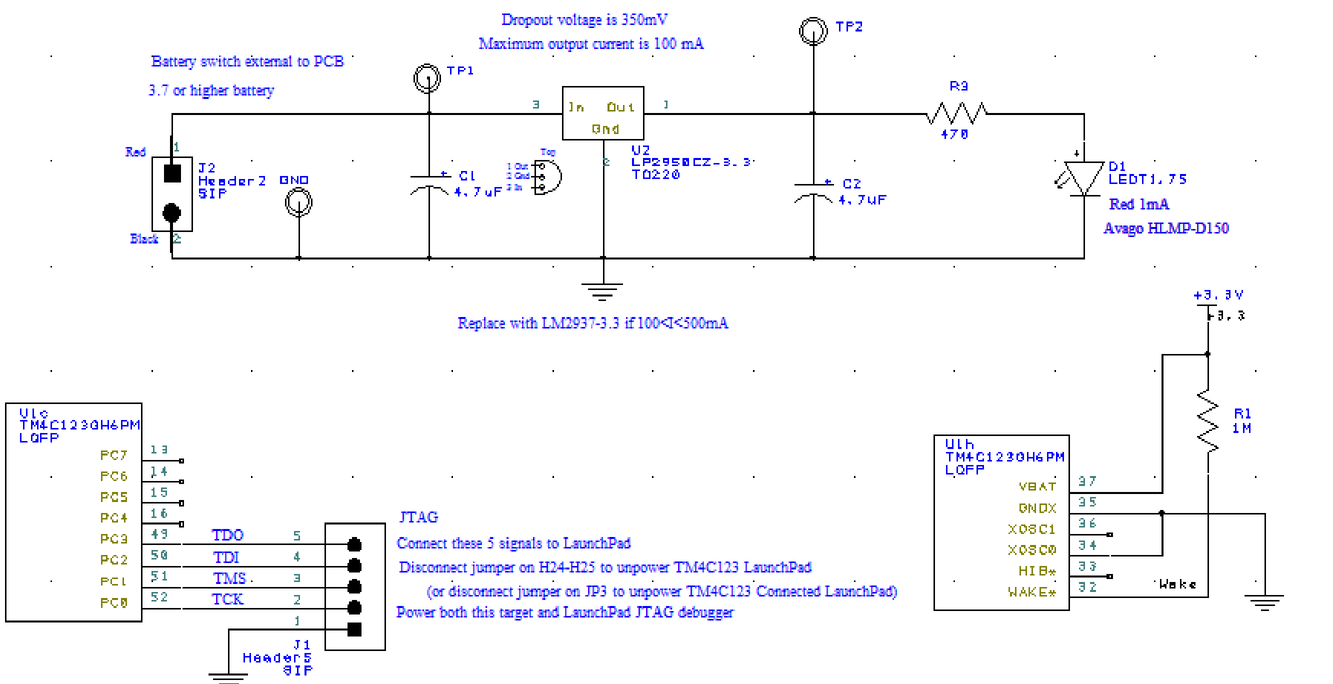


Figure 2.10: *Details for our Regulator and JTAG Debugger circuits. The JTAG will allow us to flash our software onto our PCB board. The regulator provides a stable output voltage of 3.3V from the unregulated power supply input.*

Cardboard mockup of the PCB layout

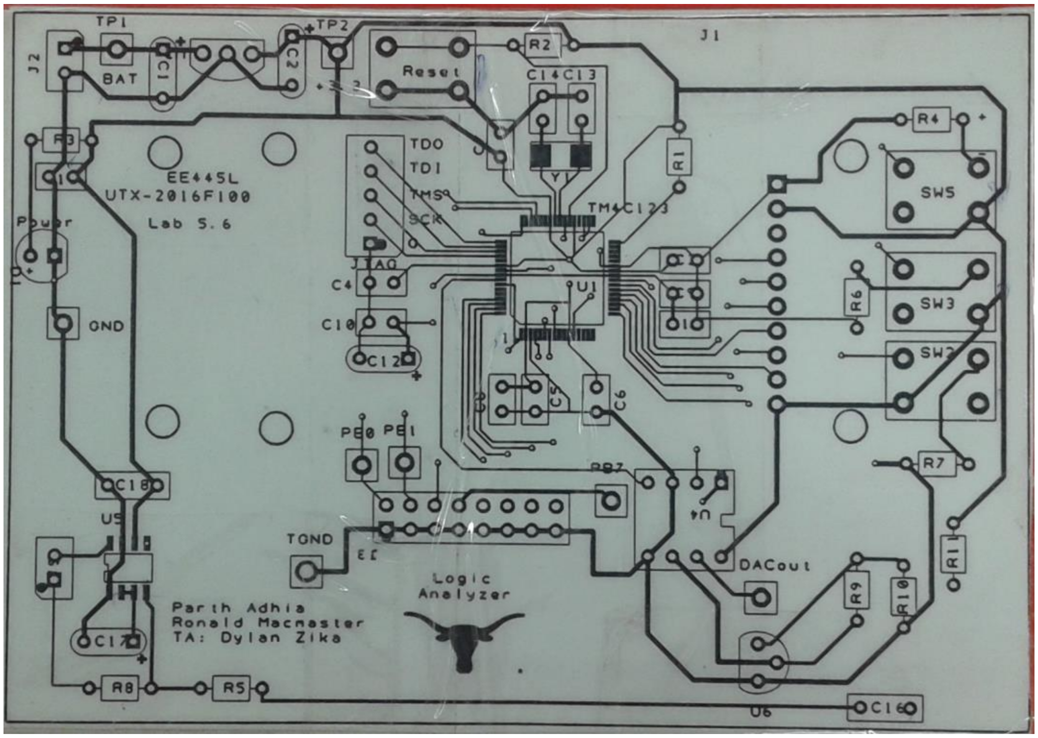


Figure 2.11: *Top Copper and silk layer of our PCB board. This is the front of our cardboard layout.*

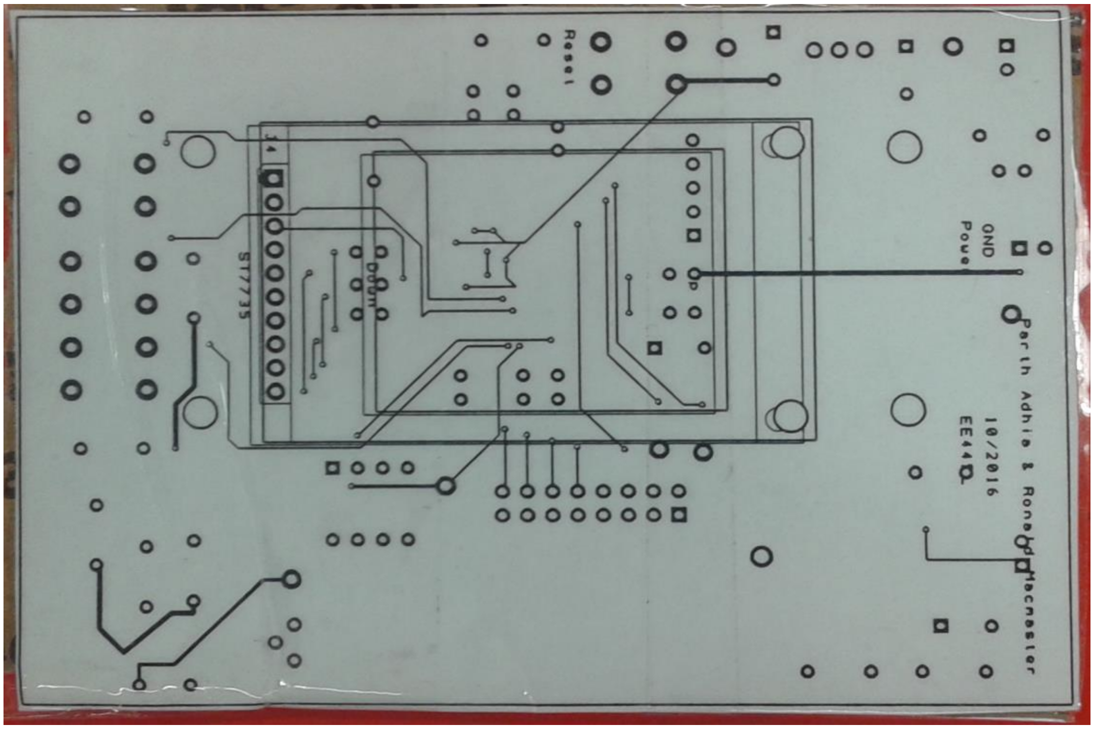


Figure 2.12: *Bottom Copper and silk layer of our PCB board. This is the bottom of our cardboard layout.*

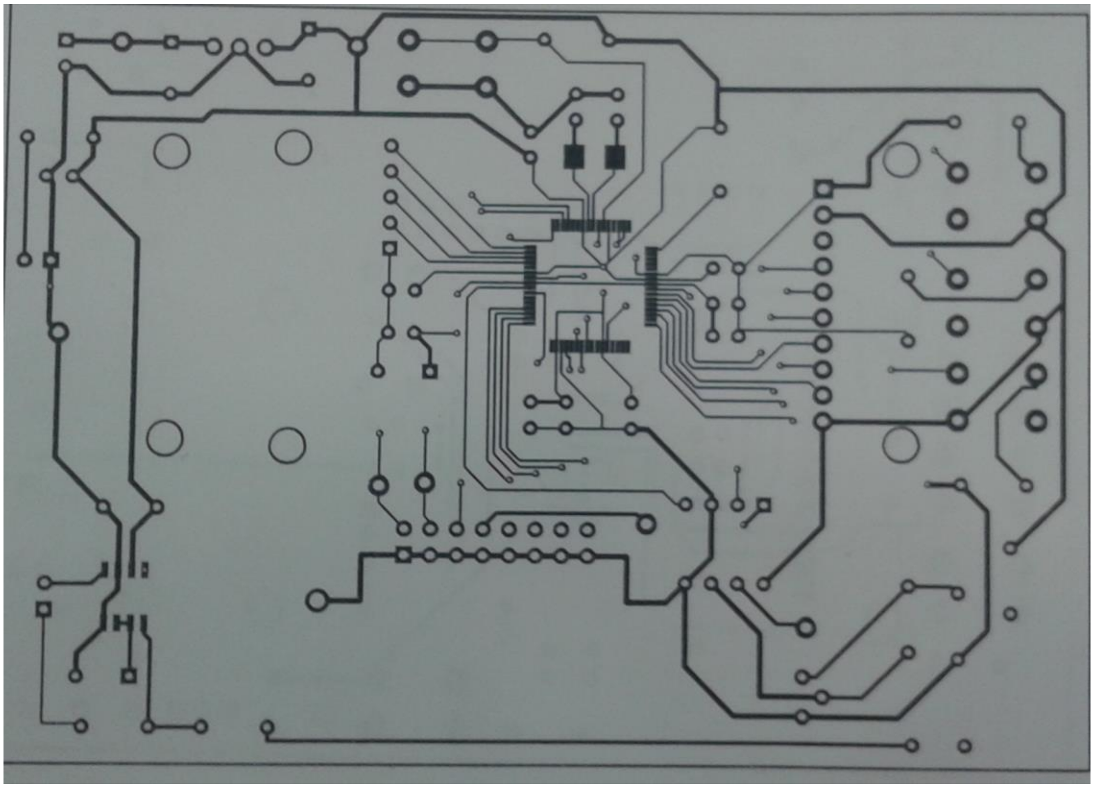


Figure 2.13: *Top Copper ONLY layout of our PCB board. This can be used for grading*

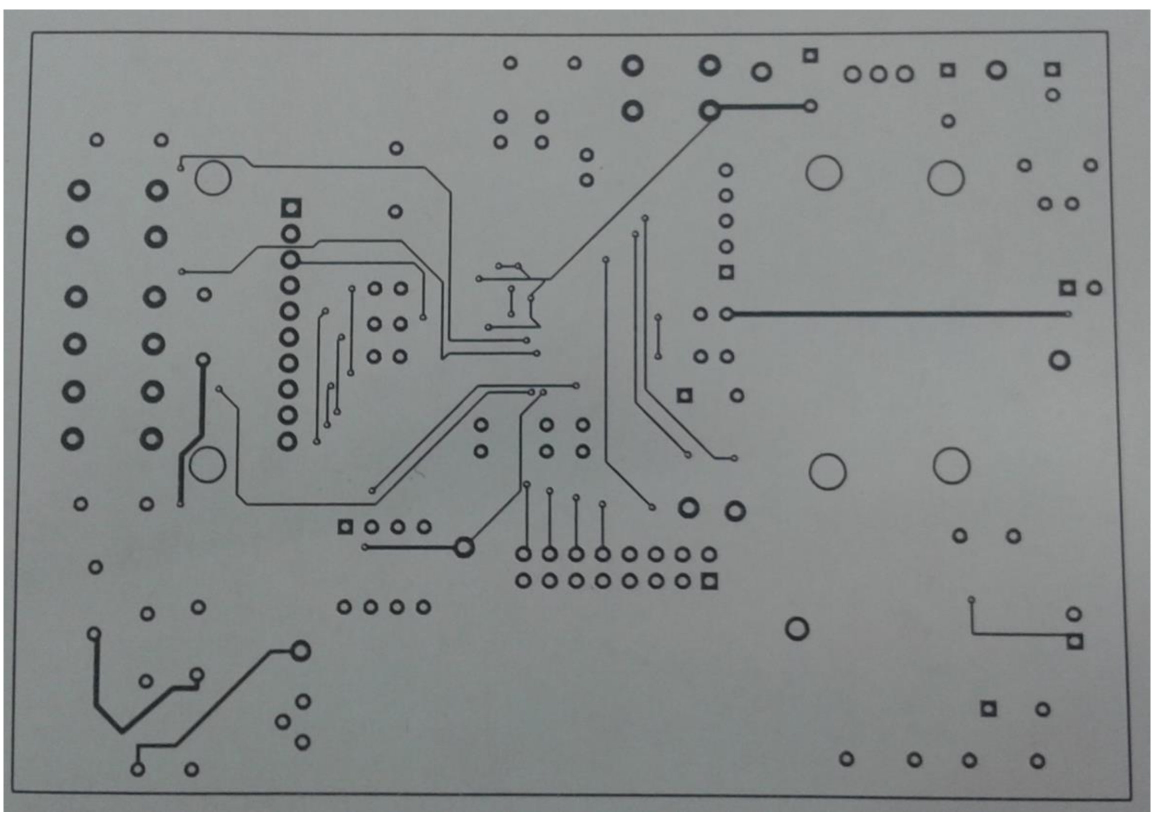


Figure 2.14: *Bottom Copper ONLY layout of our PCB board. This can be used for grading*

3.0 Measurement Data:

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

Bill of Materials			10/10/16
EE445L Fall 2016			EE445L pays for the PCB
Total Cost:		**$73.23**	
Quantity	REF DES		Description
1	J4		
2	C7 C9	ASM	32-ohm speaker
7	C15 C4 C6 C10 C11 C16 C18	CAP	Ceramic, Z5U, -20/+80%, 0.47 uF
3	C8 C3 C5	CAP	Ceramic, Z5U, -20/+80%, 0.1 uF
2	C13 C14	CAP	Ceramic, Z5U, -20/+80%, 0.01 uF
1	C12	CAP	Ceramic C0G, 10%, 10 pF
3	C1 C2 C17	CAP-Tant	Tantalum, 16V, 20% 2.2 uF
8	GND TP2 TP1 PB7 PB0 PB1 DACout TGND	CAP-Tant	Tantalum, 16V, 10% 4.7 uF
1	J3	CON	Test point, black
1	J1	CON	Logic Analyzer connector
1	U1	CON	JTAG 1 by 5 male header
1	Y1	CPU	TM4C123GH6PM 64-pin LQFP
1	U2	CRYS	NX5032GA-16.000000
1	U4	IC	LM2937 3.3 V regulator TO220, 500mA
1	U5	IC	TLV5616 12-bit DAC
1	U6	IC	TPA731 audio amp (pin 1 low) 700mW
1	J5	IC	LM4041CILPR shunt diode reference
1	D1	LCD	18-bit color 128*160 1.8" TFT LCD display
1	R3	LED	Green 2mA 5mm diffused
6	R4 R6 R7 R5 R8 R2	RES	Carbon 1/6W, 5%, 470
2	R10 R11	RES	Carbon 1/6W, 5%, 10K
1	R9	RES	Carbon 1/6W, 5%, 22K
1	R1	RES	Carbon 1/6W, 5%, 100K
4	Reset SW2 SW3 SW5	RES	Carbon 1/6W, 5%, 1M
2	J2	SW	B3F tactile push button switch
1		BATTERY	Tenergy 3.7V 2600mAh Lithium-Ion 18650 Flat Top Rechargeable Battery
1		BOX	SOD6747-2.0 Kit box 6.74 x 4.77 x 2.18 in. with PCB 5.66 x 4.33 in.

Product Number, Cost, Current, and Schematic Name.

P/N	Unit cost	Cost	**Current**	PCB Artist
SK-230	$0.50	$0.50	**41**	Speaker
399-4309-ND	$0.40	$0.80	**0**	Ceramic
399-4151-ND	$0.05	$0.32	**0**	Ceramic
399-4188-ND	$0.40	$1.20	**0**	Ceramic
490-8629-ND	$0.31	$0.62	**0**	Ceramic
581-TAP225M016CCS	$0.29	$0.29	**0**	tantalum
94035	$0.22	$0.66	**0**	tantalum
36-5001K-ND	$0.23	$1.84	**0**	testpoint
609-3344-ND	$0.27	$0.27	**0**	LogicAnalyzer
	$0.29	$0.29	**0**	JTAG
595-TM4C123GH6PMI	$11.55	$11.55	**131**	TM4C123GH6PM
644-1037-1-ND	$0.51	$0.51	**0**	XTAL/NX5032
LM2937ET-3.3-ND	$1.17	$1.17	**0**	LM2937-3.3
TLV5616CP		$0.00	**0**	TLV5616
TPA731D	$1.84	$1.84	**0**	TPA731
LM4041CILPR	$0.90	$0.90	**0**	LM4041
358	$19.96	$19.96	**90**	ST7735
516-1327-ND	$0.29	$0.29	**0**	LEDT1.75
470EBK-ND	$0.02	$0.02	**0**	0.125Wresistor
10KEBK-ND	$0.02	$0.14	**0**	0.125Wresistor
22KEBK-ND	$0.02	$0.05	**0**	0.125Wresistor
100KEBK-ND	$0.02	$0.02	**0**	0.125Wresistor
1.0MEBK-ND	$0.02	$0.02	**0**	0.125Wresistor
SW405-ND	$0.17	$0.68	**0**	B3F-1050
MH48285	$7.40	$14.80	**131**	Header
SOD6747-2.0	$14.48	$14.48	**0**	Box

Explain how you chose the battery

Our music player required either a 3.7V Li-ion battery or 4.7 NiMH battery. We needed a battery that could drive the input to a 3.3V regulator for system interface. The battery also needed to power the TPA731 from 4.8 / 7.4V. We chose the ***Tenergy 3.7V 2600mAh Lithium-Ion 18650 Flat Top Rechargeable Batteries (MH48285)*** with the intention of combining two in series to provide the 7.4V power supply for our TPA731 amplifier.

First, we formulated a power budget. Our system requires a 24 hour lifetime, so we chose the battery such that $E_{battery}$ / t_{life} was greater than our average system current I_{avg}. The music player required 131mA of current to run in the worst case, so we needed a battery with at least 3144 mAh of energy supply. The batteries we selected provide a 2600mAh energy supply to our circuit which implies a 20 hour lifetime in the worst case. We decided that these batteries would suffice given the system doesn't always run at max current, and it powers an optional LCD screen. We could have met the 24 hour lifetime requirement by removing the LCD from our system.

4.0 Analysis and Discussion:

Testing Procedure Overview

Eight pins are attached to the top portion of the logic analyzer connector and can be used to test the PCB board. The pins connected to the logic analyzer (in order) are PB0, PB1, PB3, PB7, PD0, PD1, PD2, PD3. PD0-3 are attached to the logic analyzer to increase the redundancy of the testing system. If Port B or Port E fails, a jumper cable can reroute connections to available replacement pins on port D. Any test signal can be echoed on port D and examined using a logic analyzer via the 16 pin connector.

It is recommended that testing be carried out on the Port B pins attached to the logic analyzer connector. The JTAG module is used to download the software onto the board. Throughout system operation, a digital sample between 0 and 4095 is forwarded to the DAC through the PB7 (via SSI protocol). It is recommended to connect a logic analyzer to the connector and look at the DAC SSI input waveforms.

Recommended Testing Procedure:

1. Flash the software onto the board using the JTAG module. Run TM4C123 software that generates a 1 kHz sine wave on the DAC with a desired amplitude.
2. Connect an 8-channel logic analyzer to 16-pin header, J3.
3. Compare the waveform with expected input to the DAC. (Checks for software / SSI Interfacing errors). This step ensures the functionality of data transfer from the pin PB7 to the DAC is correct.
4. Test points are placed at various places across the board where an oscilloscope probe can be used to measure analog signals. These include the DAC Output, the input to the TPA731 Amplifier (DacPostCap), and two battery testpoints near the power regulator TP1 and TP2.
5. Connect an oscilloscope to DACOut and measure the output DAC voltage for 8-16 samples. Confirm that the DAC has a linear, monotonically increasing response to its digital input samples.
6. Connect an oscilloscope to TP1 and ensure the battery provides a supply voltage close to 3.7 volts. Then, confirm the voltage at TP2 is a steady 3.3V.