ECSE 421 – Embedded Systems

System Design Document

Team 1:
Walid Hussein (260215780)
Samuel Cormier-Iijima (260174995)
Amin Mirzaee (260209556)
Benjamin Nahill (260174420)
Jules Farajallah (260188388)
Bassel Khaddam (260215343)
Christopher Bedirian (260260217)
Dong Kwak (260244875)

March 11, 2010

Contents

List of Figures

Purpose of This Document

This System Design Document (SDD) provides the reader with a general, yet complete description of an embedded digital audio amplification device. This includes all of the hardware components necessary, features and capabilities of these hardware components, and connections between each of these devices. This document will also describe the software components necessary, including operating system requirements, timing requirements, deadlines, program states, and interactions points between the user and the software based system. The design of the system which is described herein is a manifestation of the general requirements and deadlines listed in the System Requirements Specification (SRS), with some considerations into production feasibility and intuitive human interaction.

Chapter 1

Overview

The complete system consists of three main layers. The highest level is the user interaction layer, which consists of the user interface through which the system is controlled. This user interface will provide the user with feedback as to the current system state, as well as allow for modification of the systems state through a simple menu system. The middle level in this embedded system is the software layer, wherein the current system state is stored. This layer acts as a mediation layer between the user interaction layer and hardware layer, by controlling the relatively complicated inputs to the hardware based on a set of simple options specified by the user in the top layer. This layer will consist of a simple set of functions built on top of a real-time operating system known as Free RTOS. It will also poll for the hardware state at set intervals (using hardware timers) and check for error conditions. The lowest layer in the system is the hardware layer, which controls the amplification of sound as per the users preferences. This layer is the most important and complicated layer of the system and will consist of the hardware connections between various separate hardware devices. This hardware interacts with the input signals both for amplification purposes as well as audio sampling and state feedback to the software layer. It also hosts and executes the software and provides it with timers so as to allow it to function.

Chapter 2

Hardware

2.1 Controller: Microchip PIC32MX695F512H

The core of the system is the powerful Microchip PIC32MX695F512H controller, which directly interfaces with each of the system's peripherals. This controller was chosen due to the high processing speed, low cost, and abundant peripherals. It operates at 3.3V with an instruction clock of 80MHz, referenced to an 8MHz crystal for improved stability with the extreme temperature fluctuations provided by the amplifiers. The controller is based on a MIPS32 M4K 32-bit core, featuring a 5-stage pipeline, 512kB of flash program memory (with an additional 12kB available for a bootloader), and 128kB of RAM. The chip uses a a 64-pin TQFP package.

2.1.1 Hardware Peripheral Support

The PIC32MX695F512H Its abundant hardware peripherals include:

- 4 I^2C channels
- 3 SPI channels
- 16 channel 10-bit ADC
- 2 analog comparators
- 10/100Mbit ethernet controller
- USB 2.0 transciever with On-The-Go (OTG) and full-speed capabilities
- 6 UART channels
- 5 16-bit timers
- 5 Output Compare modules
- 5 Input Capture modules

- Up to 53 general purpose I/Os, many of which are 5V tolerant.

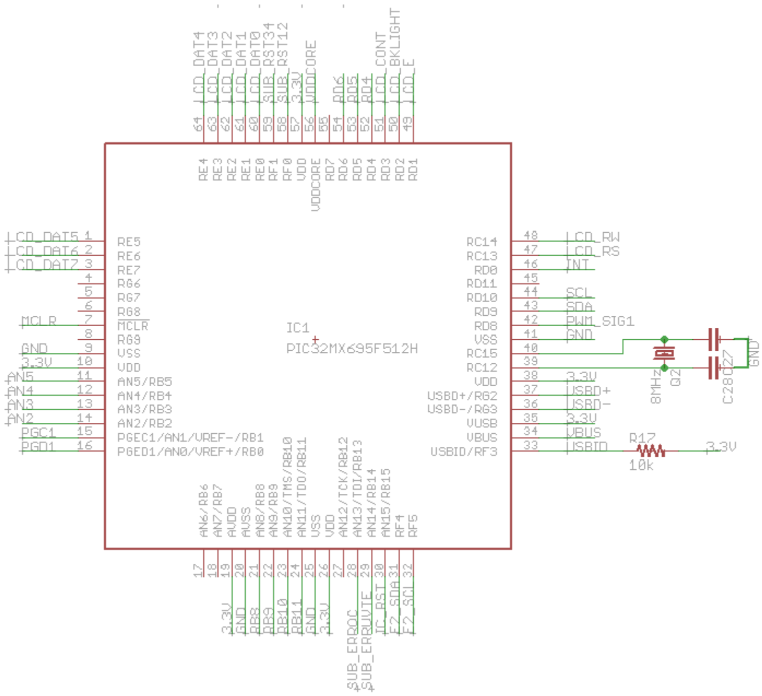

Figure 2.1: Schematic of the PIC32MX695F512H and surrounding connectivity

The following communications peripherals are used in our system:

I^{2}C

I^{2}C (**I**nter-**I**ntegrated **C**ircuit) is a bi-directional, synchronous, serial bus which requires minimal connectivity and can have any number of masters or slaves. Only two signalling wires are required: SDA (**S**erial **DA**ta) and SCL (**S**erial **CL**ock.) Since, unlike with SPI, there is no *select* line, slave selection is done by prefixing all operations with the address of the intended slave. The I^{2}C bus is used primarily as a messaging bus due to its minial connectivity requirements and low maximum clock rate, which is usually 100 or 400kHz, making it ill suited for applications requiring high data throughput. All of our peripherals operate only as slaves and the controller acts only as a master.

10-bit ADC

The 10-bit **S**uccessive **A**pproximation **R**egister (SAR) **A**nalog-to-**D**igital **C**onverter (ADC) has a 16-input analog multiplexer and can sample values from 0V to 3.3V (V_{DD}) at up to 1MHz.

USB Transciever

The USB transciever is not used in the system for any essential function. It may be used for programming if there is any trouble with the PICkit2 programmer. The Microchip USB library provides facilities to implement the USB **C**ommunications **D**evice **C**lass (CDC) for simple messaging with a PC for easy debugging.

General Purpose Input/Outputs

The selectably bidirectional I/O banks are robust, offering a number of features that are configurable on a per-pin basis:

- Weak pull-ups for inputs to eliminate the need for external resistors
- Generate interrupt on selectable edge
- Select complementary or open-drain to allow driving voltages higher than V_{DD}

Output Compare

The output compare modules compare the value of a timer to one or two fixed values and with output connected to an I/O pin. For our applications, this allows the ability to output a PWM signal using a timer that is continuously looping past a single comparison value.

Input Capture

Input capture allows timer values to be captured upon external events. This is useful in a wide array of applications where pulses need to be detected and characterized. Our system uses it to take input from a PWM signal.

2.1.2 Programming

Microchip offers a free C compiler, MPLAB C32, for educational use. Compiled code can be loaded to the chip using the **I**n-**C**ircuit **S**erial **P**rogramming (ICSP) header on the board and a PICKit2, which is a small USB programming device. Programming may also be done directly over USB to the controller by incorporating Microchip's USB bootloader into our code.

2.2 Amplifier A – Cirrus CS4245

The Cirrus CS4245 is a 4-channel Class-D (digital) amplifier with an integrated stereo ADC and audio signal processing capabilities.

2.2.1 ADC

The 24-bit 48kHz SAR ADC connects to the output of the input bandpass filter. This bandpass filter eliminates higher frequency noise as well as isolating the $V_{DD}/2$ DC bias applied internally to center the input in the ADC range. This high-frequency noise is particularly dangerous, as it can include high voltage spikes that can damage the ADC. Additionally, it is detrimental to audio fidelity to sample at above the Nyquist freqency ($f_s/2$ = 24kHz), as components above that frequency will appear as if they were folded over the Nyquist frequency.

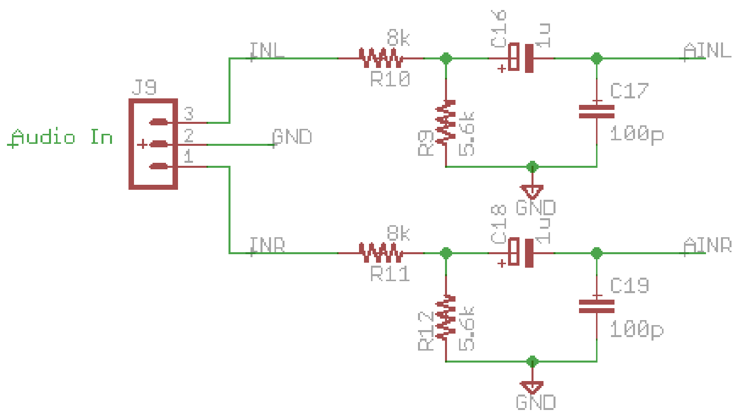

Figure 2.2: Schematic of the amplifier input filters

To operate the ADC, the system uses a 24.576MHz (48kHZ*512) crystal for its clock source.

2.2.2 Output

The output of a digital amplifier is a PWM signal which must be filtered through an analog network to reproduce the input signal. The frequency of this signal is configurable on the amplifier (to reduce AM interference) and is normally in the range of 300 to 400kHz. Logic-level outputs are also provided to allow the signals to be passed to another amplifier. The CS4525 can be configured to mix a subwoofer channel as well, using one of these logic-level ouputs.

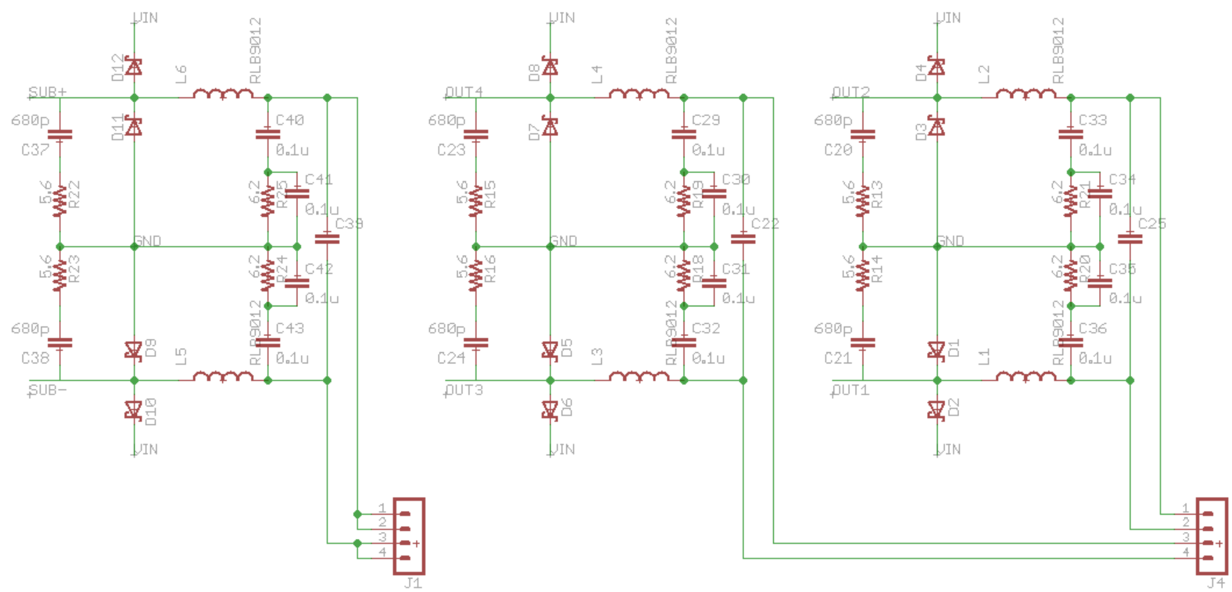

Figure 2.3: Schematic of the output filters for the subwoofer, left, and right channels

Though there are 4 single-ended outputs, we use them in a stereo **B**ridge-**T**ied **L**oad (BTL, also known as *H-Bridge*) configuration which uses two single-ended outputs, one inverted from the other, for each channel. The difference between each pair is taken as the output for each channel, providing a maximum voltage swing of $2V_{IN}$. For a fixed load impedance, this is an 4-fold increase in power over a single-ended configuration.

2.2.3 Communications

The CS4525 is controlled over I^2C in addition to asynchronous interrupt (output) and reset (input) lines. The LSB of the slave address of the device is configured in hardware by tying a pin to either V_{CC} or ground. The amplifier cannot report errors solely using I^2C, since it is required that the master initialize each communication. The asynchronous interrupt line is therefore used to signal to the controller that it must read the *Interrupt Status* register in order to determine the nature of the error. The accepted logic level of the amplifier is configurable and set to 3.3V in our case.

2.2.4 DSP

The onboard DSP of the CS4525 provides a number of built-in effects that can be individually selected and controlled:

Equalizer
: Adjustable gain for configurable frequency ranges corresponding to *Bass*, *Mid-range*, and *Treble*.

Dynamic Loudness Compensation
: The input volume is scaled to maintain an acceptable short-time average power. This has the effect of normalizing volume levels to improve audibility of material.

Thermal Foldback
: In the case of an unacceptable increase in temperature, the amplifier can automatically reduce its volume incrementally until a safe temeratureis reached. This should offer the protection necessary to avoid the need for shutdown due to a thermal error condition.

BiQuad Filter
: A standard biquad filter is available with programmable filter coefficients to implement a generic, user-designed filter.

2.2.5 Power

As a class-D amplifier, the CS4525 is very efficient, delivering approximately 90% of its consumed power to the loads. It uses a compact 48-pin QFN package with an additional thermal pad on the underside of the chip for heat transfer. The chip is designed to operate without a heatsink, therefore imposing design restrictions on the circuit board to maximize distribution and dissipation of heat.

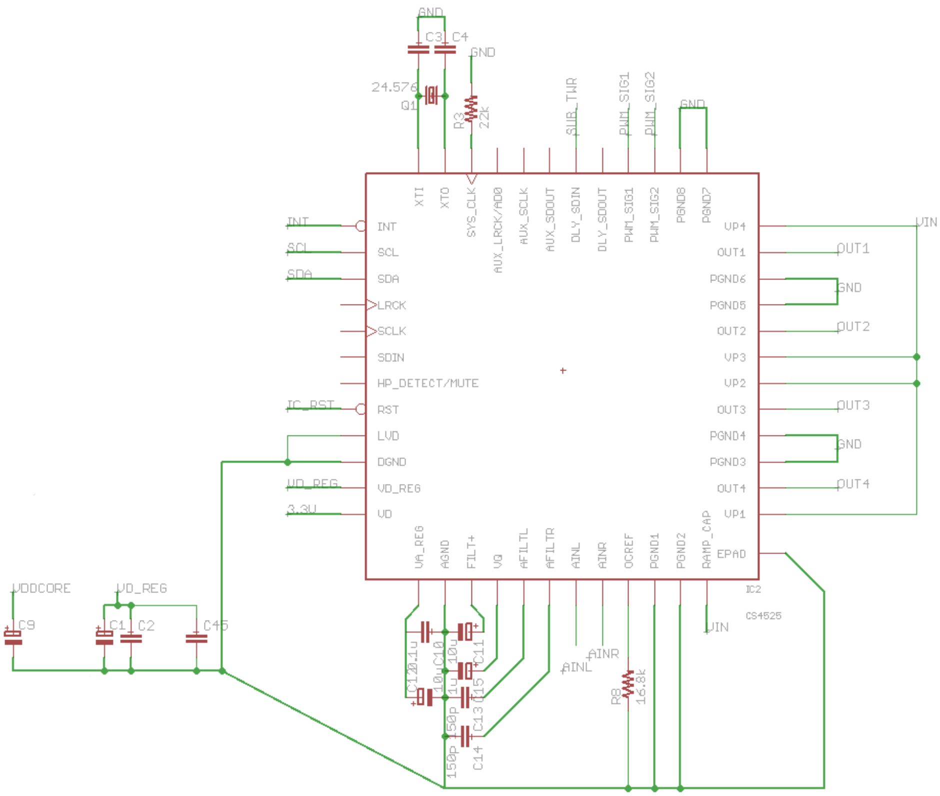

Figure 2.4: Schematic of the CS4525 and its connectivity

2.3 Amplifier B – Cirrus CS4412

The Cirrus CS4412 contains the same amplifier circuitry as the CS4525 but without the ADC and signal processing features. It is therefore well suited as a peripheral to the CS4525, amplifying an additional channel. It uses a remixed, processed, logic-level signal from the CS4525 for input and is connected in a mono BTL configuration to a lower impedance subwoofer. As this amplifier is not capable of any bus communications, any errors are caught by the interrupts on the PIC32. For thermal conditions that would not warrant an error but do require notification of the amplifier for the application of thermal foldback, there is an additional line to notify the CS4525 of this state.

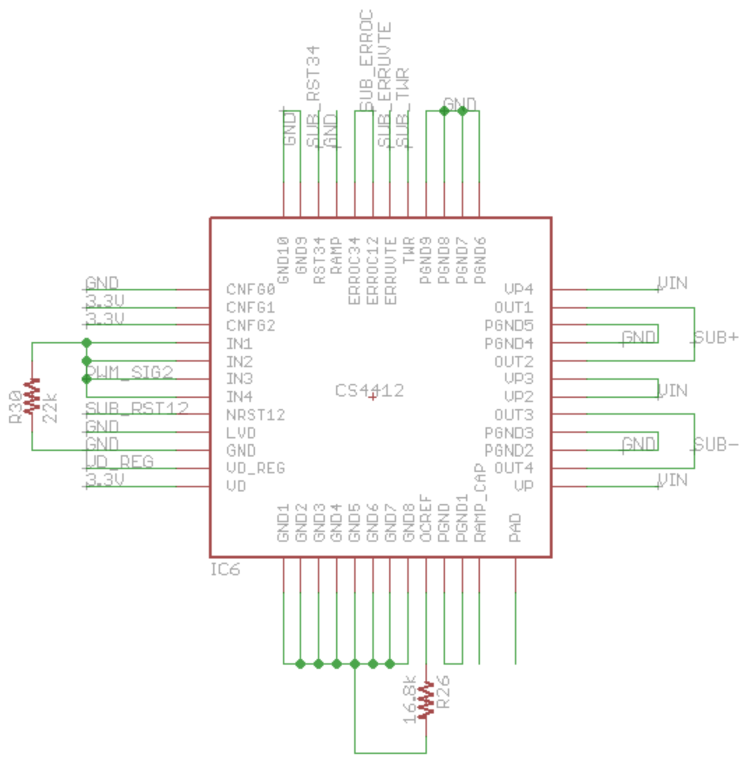

Figure 2.5: Schematic of the CS4412 amplifier and its connectivity

Like the CS4525, it can handle up to 30W but is driving only a single load. The CS4412 also comes in a 48-pin QFN with a similar thermal pad on the bottom, imposing the same PCB layout restrictions.

2.4 EEPROM – Microchip 24FC1025

The 24FC1025 is a common I^{2}C EEPROM addressing 1024kB of data and offering a minimum of 1 million write cycles and 200 years of data retention. As the microcontroller has only flash (which supports far fewer write cycles) available as non-volatile memory, the system uses the EEPROM for long term storage of settings. As there are an excess of I^{2}C channels, the system uses a second channel rather than sharing with the CS4525. The chip comes in a n 8-pin PDIP and happily operates at 3.3V.

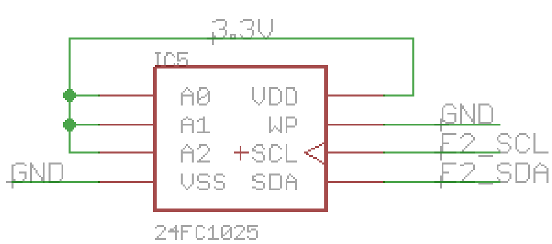

Figure 2.6: Schematic of the Microchip 24FC1025 EEPROM and its connectivity

2.5 LCD – Hitachi HD44780

The 16x2 blue-backlit character LCD employs the industry-standard protocol of the Hitachi HD44780 controller. The specifications for the protocol allow for either a 4-bit or 8-bit wide data bus, and due to the large number of unused pins, we have opted for the faster 8-bit bus. The 16-pin HD44780 header includes these 8 data pins as well as *Enable*, *Read/Write Select*, and *Register Select*. Additionally, there is a connection for power to the backlight as well as a contrast adjustment pin usually controlled with a voltage divider. These are both instead controlled by PWM, so as to be software-adjustable. As the LCD is natively 5V, the pins on the controller must be used as open-drain to allow access to the full 5V of the LCD. The LCD is accepting of the 3.3V input provided by the microcontroller. Additionally, the microcontroller pins used for the data lines from the LCD are tolerant of the 5V from the LCD.

2.6 Buttons

Capacitive buttons are implemented using the PIC32MX695F512H's ADC, which is used periodically for each of 4 buttons: *Menu/Select*, *Exit*, *Home*, and *Mute*. Capacitive buttons use the variations in capacitance of a conductor caused by the user's finger touching the button. To do this with an ADC, the ADC is first connected to V_{DD} by the multiplexer to fully charge the sample-and-hold capacitor in the ADC. The pin for the button is set as a digital output to ground to ensure that it holds no charge. The pin is then selected as an analog input, connecting the uncharged capacitance of the switch to the sample and hold capacitor in the ADC. This has the effect of a capacitive voltage divider, presenting an approximate relative value of the capacitance of the switch. The system tracks those values over time, watching for variations from an average of recent samples.

2.7 Continuous Rotary Switch

The rotary switch uses a 12-position make-before-break rotary switch and 3 general purpose I/Os to detect each transition. The selector pin of the rotary switch is tied to ground, with 3 input pins assigned in cycles around the switch with weak pull-ups enabled. At any rest state, there will be one pin pulled to ground by the selector. In the transition state in either direction, one of the adjacent pins will throw an interrupt on the falling edge as it is pulled to ground by the selector. Only when the previous pin becomes disconnected will the switch operation be recognized as complete. Since the total number of positions is a multiple of 3, the order of the pins on the switch will be repeated 4 times.

2.8 Power Regulation

The board requires 3 stable voltage levels to operate:

3.3V (V_{DD})

3.3V is considered the primary logic-level voltage for the system, powering the amplifier logic, the microcontroller, and the EEPROM. This level is provided by an adjustable LM317 positive regulator with appropriate component values to provide 3.3V. This regulator can deliver up to 1.5A, which is well above the maximum power draw of dependent components.

5V

5V is required for the LCD and is provided by an LM78L05 5V positive low-dropout regulator. This regulator is able to provide up to 100mA, well exceeding the maximum power draw of the LCD.

V_{IN}

This is the level provided by the power supply of approximately 18V. This powers the amplifiers and the regulators for other voltage levels. It must safely handle an estimated 80W at the very minimum.

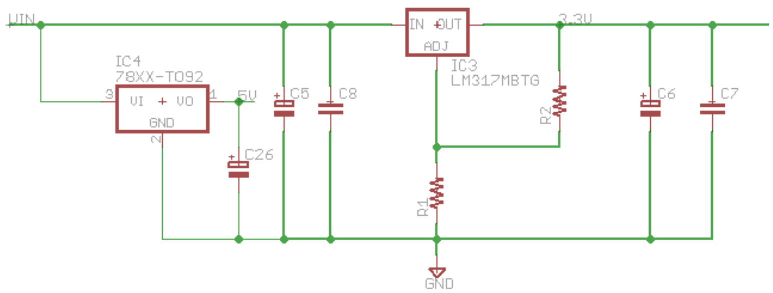

Figure 2.7: Schematic of the power regulator circuitry

2.9 Circuit Board

The circuit board is custom-designed 2-layer FR-4 PCB. It was necessary to use a custom PCB due to the complexity of the circuit, noise considerations, and number of surface mount components required. A number of extra pins are exposed for debugging and any further modification of the circuit. Layout was done with the Cadsoft EAGLE circuit design software.

The amplifiers require special consideration, as there must be an easy path for heat to leave to surrounding areas. This meant placing a number of thermal vias under each of the chips and trying to ensure that there is as extensive of a ground plane as possible in the area for the heat to travel.

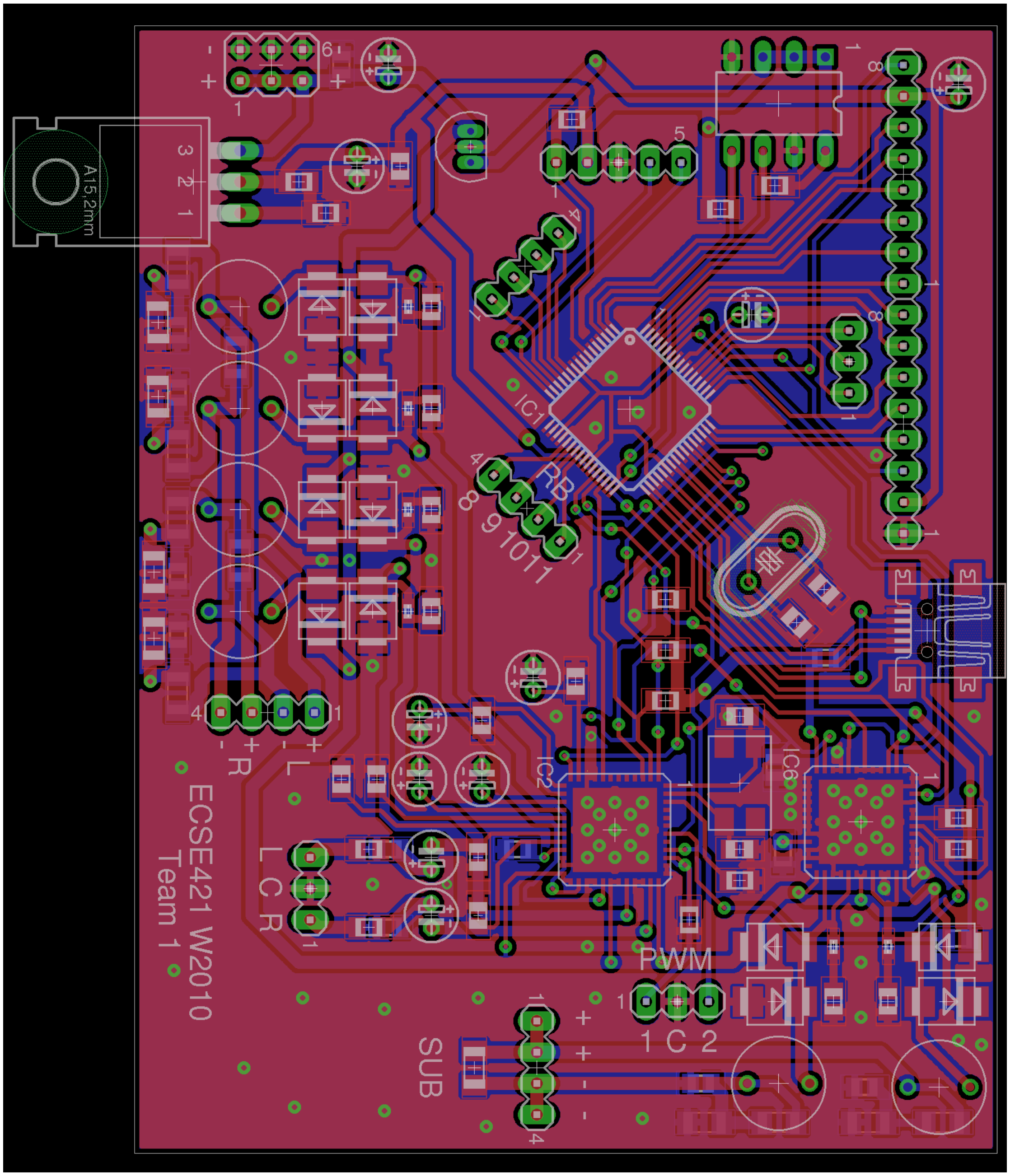

Figure 2.8: Printed circuit board layout

Chapter 3

Software

3.1 LCD Driver

The LCD driver is controlled using a list of functions that write to the LCD board in different ways. There is a function that writes a new line, a function that writes to a specified column and row, there is a function that writes a single character, a function that can write a single byte, and a function that can clear the screen so that more data can be written to it. The purpose of having all of these different ways to manipulate the LCD screen is two-fold. One, the various functions for writing to the LCD allowed for greater control over the way the user interface is designed. The functionality provided by the LCD interface allows us to implement features like writing non-ASCII characters to the screen (necessary to provide the ability to show a visualization when no music is playing). Second, we can use many of these functions during testing to ensure that the LCD board, and the modules connected to it functioned properly. Additionally the brightness and contrast can be controlled by using a PWM signal. There are additional software functions for controlling these variables by generating different pulse widths for each of these variables based on settings defined through the menu system. These functions together provide a driver interface for the rest of the software to use.

3.2 User Interface

One of the more important software design issues that we needed to consider was the user interface (UI). When designing any system that is user-controlled, it is crucial to design a UI that is both simple and effective. We have chosen to make the interface based on a state machine implemented in an event-driven manner, similar to many real-time systems. The structure of this state machine is described in more detail below; a diagram can be found in REFERENCE HERE.

Initially, the system is in the *Main* state. In this state, a user can control the volume using the knob, and the LCD will display the volume level. This display times out after 10 seconds.

As described in 2.6, the system also has four available buttons. In any state, the bottom button (the home button) will cause the system to return to the *Main* state. This provides the user with a quick way to exit any menu. The top left button (the mute button) will also work in every state, and will cause the system to either mute or unmute the audio output. The top middle button (the menu/select button) will cause a state-dependent transition corresponding to "selecting" a menu option or entering the main menu. The last button (the back button) will take the user to the previous menu. There will be a timeout time (which can be chosen by the user, but defaulted at 30 seconds) after which the system will go back to the initial state.

The menu structure corresponds directly to the states in the state machine. These are shown in their logical structure below.

- Equalizer
  - Bass
  - Treble
  -
  - Adaptive Loudness
- System
  - Enable/Disable Spectrum Analyzer
  - Screen Contrast
  - Screen Brightness
- Presets
  - Load Preset
    * Preset 1
    * Preset 2
    * Preset 3
  - Save Preset
    * Preset 1
    * Preset 2
    * Preset 3