Production Model Design Report

F2019 – ECE 298

|  |  |  |  |
| --- | --- | --- | --- |
| Lab Section: | 4 | Group: | 11 |

# Team Members

|  |  |  |
| --- | --- | --- |
| # | Name | Role |
| 1 | Matthew Tang | UART, Defeat Control for Common Alarm, Mux Control, RCT, LCD Display, Entry/Exit Sensors, Software, Software Filtering |
| 2 | Michael Meng | Audio Alarm, Sound Sensor, LED Indicators, Soldering PCB Parts, Hardware Filtering, Software, Pushbutton Usage |

# Design Overview

## Problem Statement

The need for Home Security has skyrocketed in the past few years as home invasion rates has increased drastically. As technology keeps advancing and burglars become smarter, Home Security is becoming a necessity for many households. Design a Home Security system that can monitor a minimum of 4 zones concurrently to detect doors and windows being opened. A sound sensor should be used in 1 of the zones paired with hardware & software filtering to reduce false tripping. If any of the armed zones detect a door or window being opened or the sound alarm detects a significant change in sound levels, an audio alarm must be played to signify that one of the zones has a problem. Additionally, there are 3 states ZONE\_OK, ZONE\_NOT\_OK, ZONE\_ARMED that are required to be displayed on the LCD and/or the LED’s. There must also be a defeat control to enable/disable the audio alarm if tripped. A computer interface may be optionally implemented to arm/disarm zones at specific times.

## Design Scope

This project solves the problem by using reed switches as sensors for doors/windows and a microphone for detecting sound to monitor 4 zones concurrently. If any of the 4 zones transition from ZONE\_OK to ZONE\_NOT\_OK an audio alarm is triggered and can only be turned off by a defeat control. A passive low pass filter was used for hardware sound filtering and a modified version of Simpson’s rule was used in the software to filter out ambient noise and reduce false tripping. Zone statuses were shown on the LCD along with the RTC and zone armed statuses were shown on the LED’s beside the reed switches.

The solution also consisted of a custom computer interface that can arm/disarm zones at specific times using the RTC, display all zone information on the computer screen, and reset the system as needed. Source code for the project is located at the following reference [1]. Furthermore, the UART and RTC functionality in the source code was inspired by information found at the following reference [2].

The assumptions that were made include:

* Only 4 zones are being monitored concurrently
* Only 1 of the zones has sound detection capability
* The only way to restart the system is through UART and the computer interface
* The only way to arm/disarm zones is through UART and the computer interface
* Once a zone transitions from ZONE\_OK to ZONE\_NOT\_OK it is stuck as that state until specifically reset by commands on the computer interface

## Project Design Requirements

1. Sound sensor detects noise and changes state to “ZONE\_NOT\_OK” if noise is above a certain threshold
2. Produce sound from Audio Piezo Buzzer if any of the zones are in the “ZONE\_NOT\_OK” state.
3. Reed Switches are used for zone detection. Zones are in “ZONE\_OK” state if Reed Switch is closed and zones are in “ZONE\_NOT\_OK” state if Reed Switch is open.
4. LCD displays zone status information (ZONE\_OK or ZONE\_NOT\_OK)
5. LCD also displays RTC Clock, toggled between zone statuses using a push button interrupt
6. Turn on corresponding LED for the individual zones if the zone is (ZONE\_ARMED) and off if the zone is (ZONE\_NOT\_ARMED)
7. UART computer interface to arm/disarm zones at specific times, display status information, and reset the system to its initial conditions (All Zones Armed and ZONE\_OK)

## System-Level Design (High-Level)

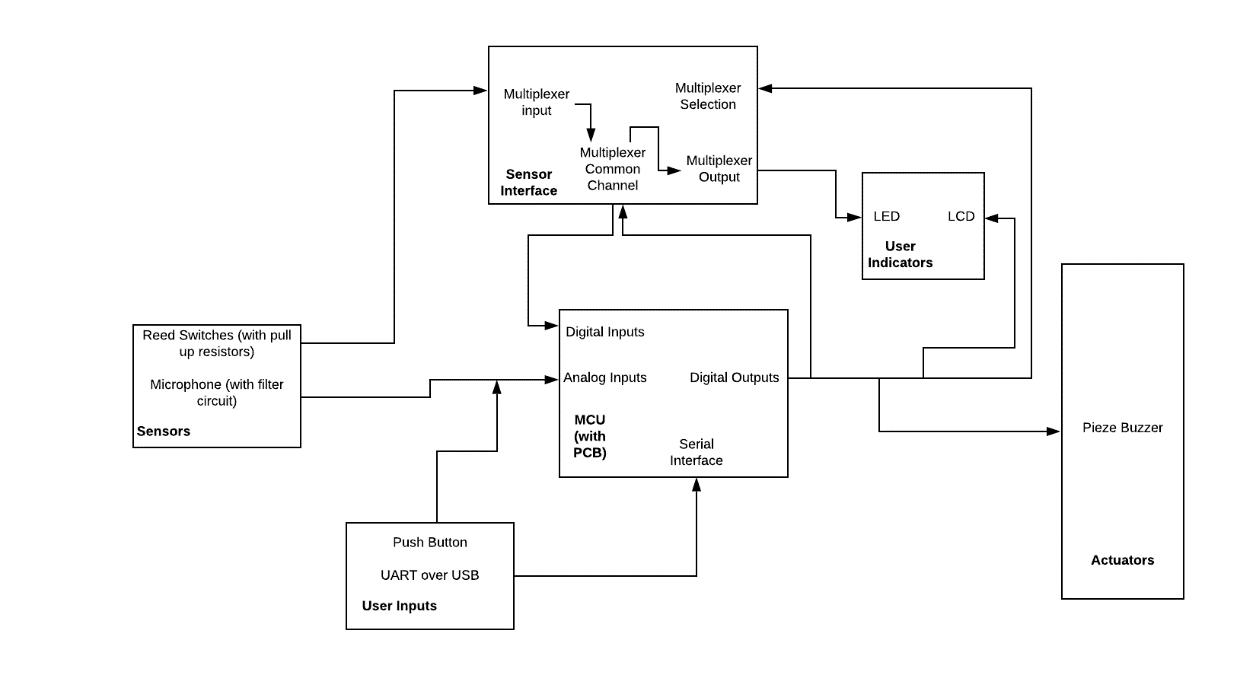


Figure 1: High Level Block Diagram of Final Prototype with PCB Breakout Board.

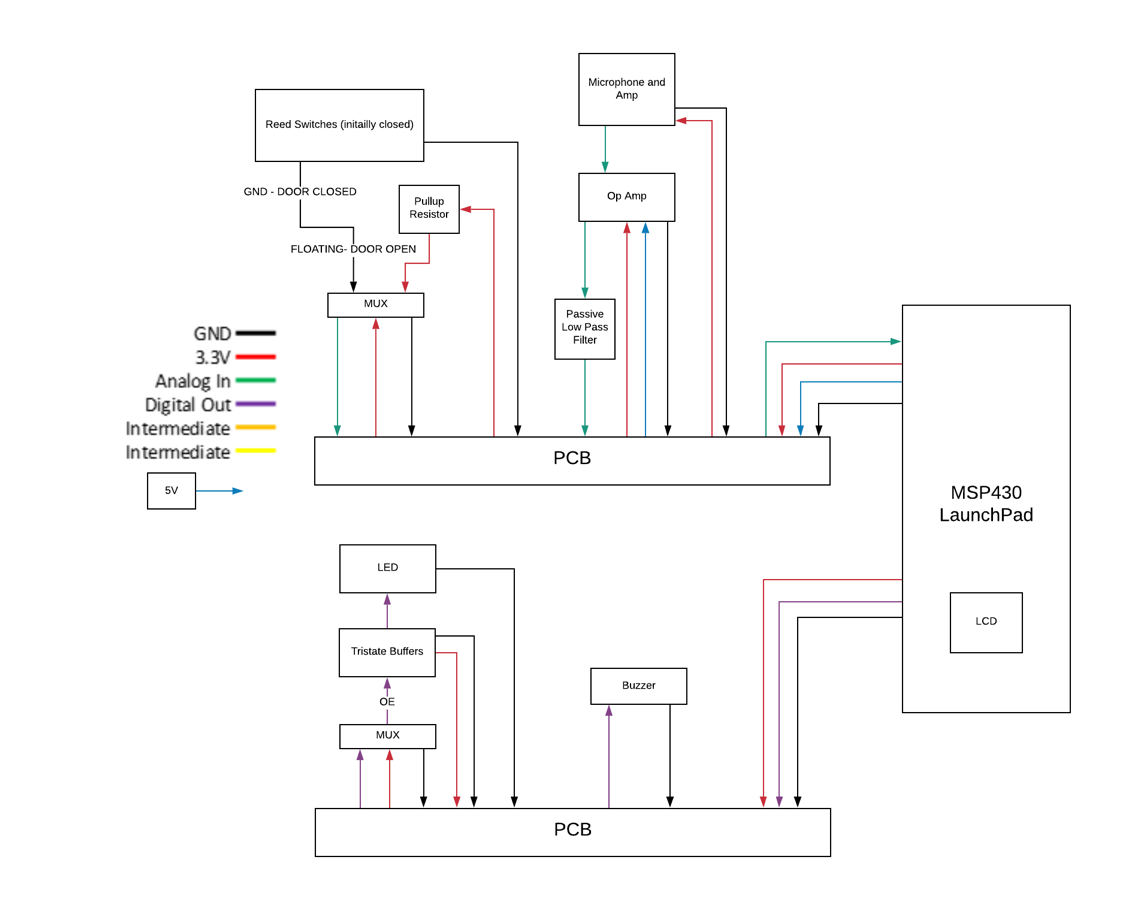


Figure 2: High Level Final Design Model Diagram with a PCB breakout Board, Added Reed Switches, MUX, and Tristate Buffers for LED’s.

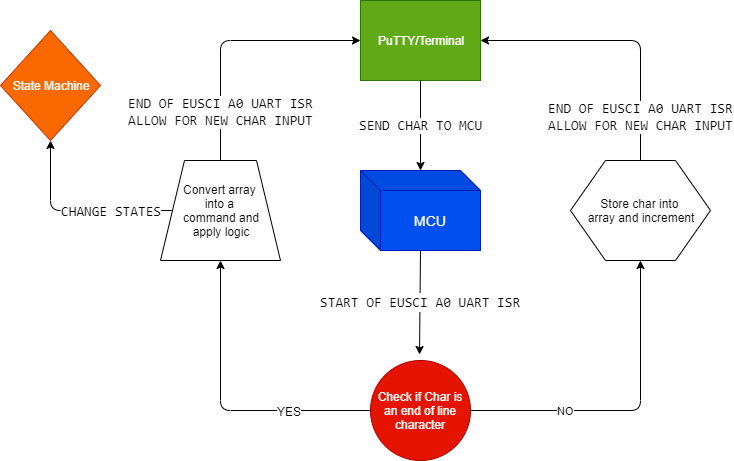
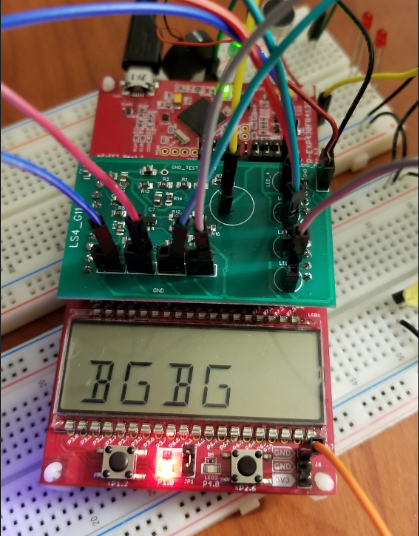
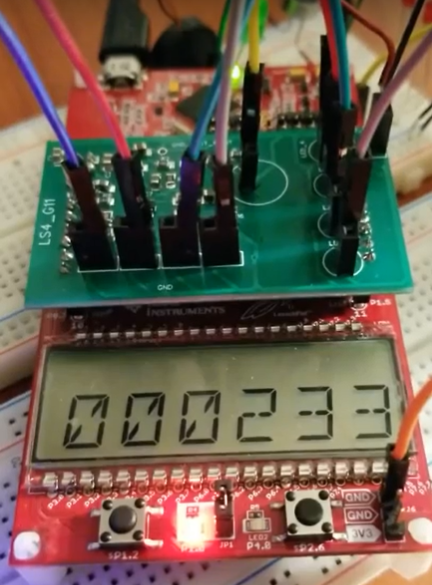


Figure 3: Flow Diagram of How UART Works and How We Can Send Commands to the MCU to ARM/DISARM Zones

## Completed Prototype

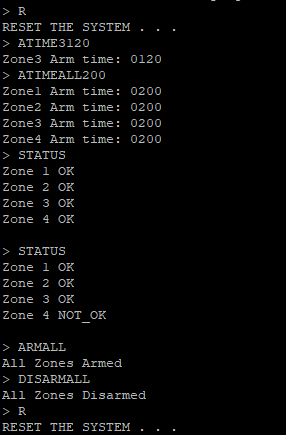
## 

*Figure 4: A top view of the completed Home Security system. Each of the four corners represent a zone, and the top right zone has a microphone for sound detection*

*Figure 6: Real Time Clock shown on the LCD*

*Figure 5: Zone statuses shown on LCD G = ZONE\_OK, B = ZONE\_NOT\_OK*



*Figure 7: UART Interface to Send Commands to MCU*

## Preliminary Production Design Changes

* For our tristate circuit with the LED’s, our power supply was going through 1 common capacitor in series before it reached the other tristate buffers. This caused some issues with trying to individually turn on or off individual LED’s, as the common capacitor was discharging the current to the other LED’s when they should have been turned off.
* There was a lack of test points on the back of the PCB which made it very hard to debug our tristate buffers and LED logic. In the future we should add more test points which will help with debugging and verifying the components work as intended.
* We should remove the tristate buffers and have a direct connection from the MUX to LED. This would save money and space on PCB in the future.
* We could improve the software filtering for the MIC. Right now, what we do is take 30 ambient noise points and calculate the average of the 30 points and set that as our new threshold. If our incoming data from the MIC is greater than the threshold, that is “a sound detected” and the buzzer goes off. This approach works sometimes on very loud claps or MIC taps. There is room for improvement here.
* We could also improve the hardware filtering. In our design, we are using a passive low pass filter. We could look to use a better filter strategy such as using a band pass filter.
* To make the project more “complete”, a suggestion would be to create CLI or add some GUI for UART to USB instead of just a terminal screen

# Member 1 Production Details

Matthew Tang – 20723384

## Design for Test (DFT)

Design for Testing is an Integrated Circuit design technique that adds testable features to a hardware and software product design [3]. The main purpose of adding these testable features is because during the manufacturing step, it allows for an easier way to validate whether the hardware contains any defects that could affect the product’s functionality. These tests are applied at various points in the hardware manufacturing flow and are executed automatically using software programs and automatic testing equipment. The most common form of DFT done in industry is based upon the Structural test paradigm [3]. Structural test is different then functional test in the sense that it does not attempt to determine if the overall functionality of the circuit is correct, instead, it will make sure that the circuit has been assembled properly all the way down to the lowest building blocks (ex: are all logic gates present) [3].

Design for Test is important to our home security system because it helps with the general assembly of the project at each stage and verifies that the model is functional without defects. When we got the PCB and started soldering our components, the main challenge was figuring out if everything we put on the board even worked. This is where DFT comes into play. If we had built in testable features (mainly hardware but software as well), at every step of our assembly we could verify that our PCB and components were working as intended. This would have made debugging our design much faster and more effective.

In our home security project, it is possible to build several test and debug features at design time. One of the most common examples would be having a debug-friendly PCB layout and test points for the different components [4]. This allows for easy readability and testability when debugging. As mentioned above in our project restrictions, the only way to reset the system is through the UART interface. This is not a good design implementation and to fix that we should add a pushbutton interrupt. This pushbutton will be able to reset the entire system to a state defined by the user. This is extremely important as we can run as many automated tests as we desire by returning the system back to its original state with just the push of a button. In addition, we could increase the observability of our home security system to allow users view the state of internal nodes. Instead of splitting the zone status and zone armed status between the LCD and LEDs, it would be much easier to pick one of the two methods (LCD most likely) and display the information there. This would increase observability and allows for a very easy scan of the entire system when multiple automated tests are performed.

A production engineer should keep these high-level set of next steps in mind to transform the prototype into a production ready model:

1. Rewire the PCB Layout so there are no long wires that span across the entire PCB.
2. Add more silk trace to the PCB for readability.
3. Add more test points near the mux and user interfacing components to help with debugging. This would be extremely useful for the LED circuit as we could test that the mux is turning on and off zones properly.
4. To fix the issue with resetting the system only with UART, it would be wise to add a pushbutton that acts as a hard reset for the entire system.
5. Use either LCD or LED for displaying zone information to improve clarity.
6. Add additional hardware logic to improve the predictability of test results, i.e. no possibility of unknowns [4]

## Energy Efficiency

Energy efficiency in electronics and integrated circuit design is very important. In simple terms, energy efficiency is doing more with less [5]. Often, efficiency is confused with effectiveness. Efficiency refers to the “wanted” effect that we are aiming for. For example, a light bulb may have 2 % efficiency in emitting light, but 98 % effective in heating up a room. We care about energy efficiency because energy consumption has skyrocketed in the fast few decades and we are in danger of using up all the Earth’s resources at the rate that we are going [5]. Energy efficiency is essential for any mass production product because we want to limit power consumption while attaining maximum value.

This topic relates to the Home Security project because if we mass produce a product that is not energy efficient to the customers, it can be detrimental. Although the energy consumed by our demo is quite low, if we were to scale up this project to use higher end devices, monitor more zones and add more technical features, the energy consumption would increase significantly.

For every zone that we add or want to extend to, more components are required, and more electrical power is used to operate the system. Currently, we are monitoring 4 zones concurrently, but if we want to monitor a house, 4 zones are not enough. We would be required to monitor more windows/doors which would require more detection components. Additionally, it is not required to display zone information using 2 different components. Technically, only 1 of the components should be used which helps to reduce the power consumption. Lastly, we power the LED’s using a tristate buffer and pull up resistors. This is not energy efficient at all. We could directly connect the LED’s to the multiplexor, remove the tristate buffers we have driving the LED’s, and that would lower our energy consumption.

A production engineer should keep these high-level set of next steps in mind to transform the prototype into a production ready model:

1. Either use LED or LCD for displaying zone information to reduce on the use of 1 component.
2. To improve our current energy efficiency of the system, when certain zones are disarmed, we could “power off” the zones so the sensors are not consuming energy when they are not needed.
3. If we are using LED to display zone information, connect the LED’s directly to the mux and remove the tristate buffers and pull up resistors we are using.
4. Reed switches are a very cheap and low energy device for detection. Unfortunately, they are not extremely accurate. This is an issue or design change that the production engineer should keep in mind. There might be a need to change to something like an IP surveillance system, or intelligent video analytics that have intrusion detection/object removal/effective storage management. These devices will increase the amount of energy being used, but it would provide a better way to detect intruders.

# Member 2 Production Details

Michael Meng – 20666979

## Design for Manufacturability

Design for Manufacturability is a crucial part of the design process. This process ensures that the design takes into consideration the downstream manufacturing processes of the final product to ensure an end product with high quality and cheap cost. DFM encompasses several key factors such as reducing costs, improving scalability, upholding high manufacturing standards, and reducing process times to increase manufacturing speed.

Design for Manufacturability is an important aspect to our design process which would help ensure our design is cost efficient, high quality, and scalable. Our home security project required us to develop a PCB with a lot of components on it. The challenge was figure out how to design our PCB with good spacing, accessibility of different components (user facing such as reed switches or LED’s and non-user facing such as our MUX), manufacturing processes, and the efficient assembly/testing of our board. All of these factors play a role in the DFM process. As a result, this helps us produce good quality PCB’s at a cheap cost with ease of mass production, testing, and scalability.

There are a few things that the production engineer needs to know about the design with regards to DFM before taking the prototype through to a production-ready product. One of the most important issues is the LED circuit that was used for the final prototype during demos. The LED circuit was quite packed and complicated. It included inverted tristate buffers that sunk current to a ground when the output enable was set low, and when the output enable was high the LED’s were powered by a pullup resistor system. After our final prototype, a design change that would make the PCB much easier to manufacture would be to remove the tristate buffers and pullup resistors, and just power the LED’s using a straight connection to the MUX. This will reduce the number of components needed for the PCB, reduce manufacturing time by removing the surface mount pads for the tristate components and test points, and improve the design overall by simplifying the system without sacrificing any core functionality (the LED’s will still function as stated in the design doc).

Another thing to keep in mind would be to ensure that tolerances are kept tight for the microphone circuit. If tolerances are kept tight in the design phase, and these tolerances are adhered to during manufacturing, this will reduce the chance of potential errors that would cause the circuit to fail. The microphone circuit is critical for detecting sound.

## Supply Chain Management

Supply chain management is the process of different corporation units (business, manufacturing, distribution) which cooperate to manage material and information flow. Supply chain management relies heavily on the communication and cooperation of these individual units to ensure that production quotas are met on time, products are produced at a high quality, and that costs of operation are kept low. An example of a supply chain could be the business and forecasting team sourcing raw materials from suppliers, having those suppliers deliver the materials to the production facility, manufacturing and testing the product made, storing those products in a warehouse, and distributing those products to consumers on a schedule. All of those different units work together to ensure this process is done with low cost and high quality.

This relates to our home security project because in the case of scaling, mass producing, and shipping our embedded system to customers, supply chain management is a crucial element we should consider to help optimize that process. For instance, in order to mass produce and ship these products out to customers on time, we need to consider the manufacturing and shipping process of the product. It is important to ensure that the manufacturing team talks with the logistics and distribution team to ensure that the products are made on time and can be packed/sent out to customers. This involves the work of both teams to communicate and coordinate with each other, and adheres to the principles of supply chain management which help optimize and reduce costs of overall operations.

*With regards to supply chain management, it is important for the production engineer to keep in mind that mass production with a high quality of standard and in a fixed time is important for the product. As a way of increasing manufacturing speed, we could introduce automation as a form of labour (testing, manufacturing, assembling). This will greatly speed up the manufacturing process, which could lead to faster shipping, reduced costs, and more overhead for any other unexpected events during the manufacturing process. However, with more automation, this introduces the possibility of reduced quality standards for the product. Using robots to assemble and test components may lead to missed quality checks which would increase operations costs. In order to combat this, there must be a change to how the engineer plans quality assurance checks, or keeps track of these metrics. This information will also need to be communicated to other parts of the supply chain (business and distribution) to ensure the entire supply chain is aware of the change, and can act accordingly to modify any of their own standards to support the change. This increases effectiveness of the supply chain and will optimize the production process in the long run.*

# References

|  |  |
| --- | --- |
| [1] | T. Instrument, "MSP430FR4133," Texas Instrument, [Online]. Available: http://www.ti.com/product/MSP430FR4133. [Accessed 19 November 2019]. |
| [2] | M. Tang, "Home Security System," 23 November 2019. [Online]. Available: https://github.com/Matt-Tang/Home-Security-System. [Accessed 02 December 2019]. |
| [3] | "Wikipedia," Wikipedia, 23 July 2019. [Online]. Available: https://en.wikipedia.org/wiki/Design\_for\_testing. [Accessed 30 November 2019]. |
| [4] | M. Horowitz, "Lecutre 14 Design for Testability," 2006. [Online]. Available: https://web.stanford.edu/class/archive/ee/ee371/ee371.1066/lectures/lect\_14.2up.pdf. [Accessed 30 November 2019]. |
| [5] | O. E. Ltd, "What is energy efficiency?," OVO Energy Ltd, [Online]. Available: https://www.ovoenergy.com/guides/energy-guides/what-is-energy-efficiency.html. [Accessed 01 December 2019]. |
| [6] | "What is Design for Manufacturing or DFM?," Versa Electronics, 15 May 2019. [Online]. Available: https://versae.com/design-for-manufacturing-dfm/. [Accessed 03 December 2019]. |
| [7] | "7 Areas of Supply Chain Management for Electronic Manufacturing," Versa Electronics, 24 September 2019. [Online]. Available: https://versae.com/scm-supply-chain-management-for-electronic-manufacturing/. [Accessed 03 December 2019]. |

# Appendix – Detailed Design

**Project Repository**

<https://github.com/Matt-Tang/Home-Security-System/tree/master/ECE_298> [2]

**Table 1:**

Table : Necessary Design Changes

|  |  |  |
| --- | --- | --- |
| # | Change | Reason/Notes |
| 1 | We ended up splitting the start and stop interrupts to map to the two switches on the msp430 board | We did this because our initial idea because after some planning we realized that we had access to the 2 switches on the board, and we ended using one switch for turning the state flag on, and the other switch for turning the state switch off. |
| 2 | We switched out the ultrasonic sensors to use magnetic reed switches instead | We realized we have limited timers from the board, and it would be hard to allocate those timers to support 4 ultrasonic sensors to read zone data. Reed switches would be a more efficient method and we would have less debugging in the end. |
| 3 | We ended up making a better implementation for the audio sensors | Initially, we didn’t use any software filtering, and operated it based off analog signals, and we fed the output of the microphone to an LED (which was really poor design decision). We ended up using the ADC API in the software kit and used that to detect from the microphone, and we sent a digital signal from another pin on the board to an LED. |

**Table 2:**

Table : Important Notes

|  |  |
| --- | --- |
| # | Note |
| 1 | Remembering how to use the boards built in Timers for measuring pulse width if we ever decide to use the ultrasonic for measuring distance |
| 2 | Setting up Audio sensor and being able to measure the frequencies being read using the digital oscilloscope. |
| 3 | Initially, we were having troubles with reading the audio signals from the microphone’s breakout board. We ended up looking into the starter code and used the ADC conversion code to read from the pin and compare to a static threshold to change the state of the alarm system |
| 4 | We spent a lot of time trying to use the signal generator to simulate the  ultrasonic sensor due to shortage of parts from Ridgidware. In the end after our prototype demo, we got access to an ultrasonic sensor and were able to use TimerA to properly read the pulse width of the echo from the sensor. (However, for our final design, we found that it would be better to use Reed Switches instead of ultrasonic sensors). |
| 5 | We initially wanted to use ultrasonic sensors to read state information of the 4 zones. However, we figured out during our schematic planning that sharing TimerA with the 4 zones may not be the most ideal situation, and a more optimal way of sensing zone information would be to use magnetic reed switches. |

**Table 3:**

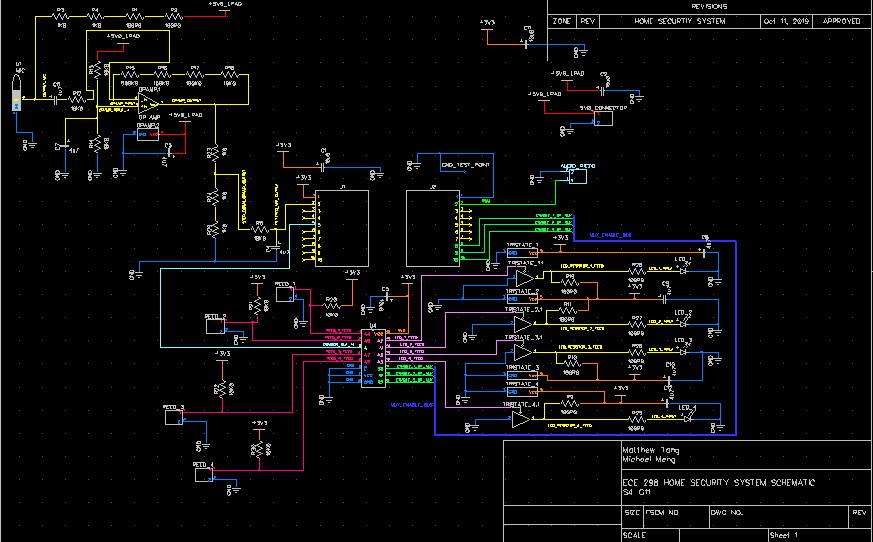
Table : Hardware Signal Test Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Signal (TP\*) | Property | Required Software Mode | Min | Nominal | Max |
| PWM Out (X)  Piezo Buzzer | Voltage | Buzzer on | 0 V |  | 5 V |
|  | Period | Buzzer on |  | 250 ms |  |
|  | Duty Cycle | Buzzer on | 0 % | 25 % | 50 % |
|  |  |  |  |  |  |
| MIC Audio\* (need test point) | Voltage | Any | 0V |  | 3.3 V |
|  |  |  |  |  |  |
| Magnetic Reed switch 1,2,3,4 | Voltage | Any | 0A |  | 3.3V |
|  | Current | Any | 0A |  | 0.3mA |
|  |  |  |  |  |  |
| MUX Common In/Out | Voltage | Any | 0V |  | 3.3V |
|  |  |  |  |  |  |
| MUX address signals S0,S1,S2 | Voltage | Any | 0V |  | 3.3V |
|  |  |  |  |  |  |
| Output LED 1,2,3,4 | Voltage | Any | 0 |  | 3.3V |
|  | Current | Any | 0 |  | 3mA |
|  |  |  |  |  |  |

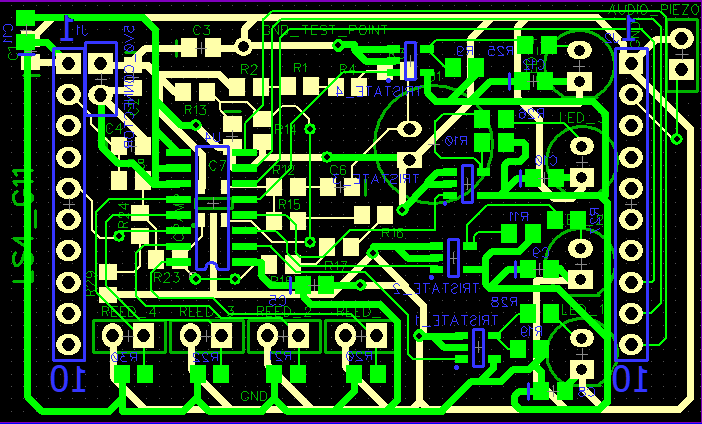
**Table 4:**

Table : Hardware Signal Connectivity

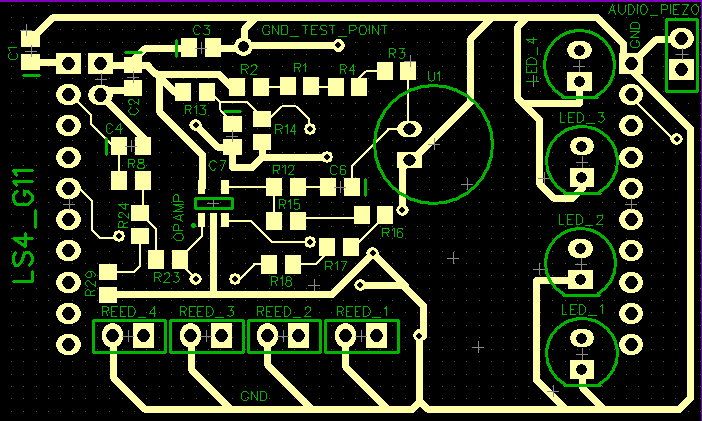
|  |  |  |  |
| --- | --- | --- | --- |
| Signal | MSP430FR4133 Pin | LaunchPad J1/J2 Pin | Prototype Connection |
| PWM Out | P1.7 (PWM) | J2 pin 19 | Piezo Electric Buzzer |
| Analog In | P8.1 | J1 pin 2 | Audio Sensor In |
| Digital Out | P1.3 (GPIO) | J2 pin 13 | Multiplexer S0 address bit (S0 used as MUX address bit 0) |
| Digital Out | P1.4 (GPIO) | J2 pin 12 | Multiplexer S1 address bit (S1 used as MUX address bit 1) |
| Digital Out | P1.5 (GPIO) | J2 pin 11­­­­ | Multiplexer S2 address bit (S2 used as MUX address bit 2) |
| Digital In/Out | P2.7 (GPIO) | J1 pin 5 | Multiplexer A control signal, will either be used for input to read reed switches, or output to turn on LED |



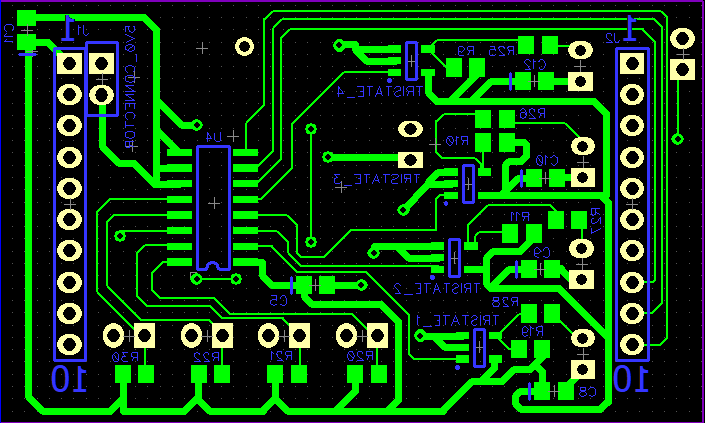
*Figure 8: Schematic Diagram Using the DipTrace Software*



*Figure 9: Top and Bottom Level View of PCB*



*Figure 10: Top Level View of PCB With User Interfacing Components on Top*



*Figure 11: Bottom Level View of PCB With Mux, Tristate Buffers, and LED Circuit on the Bottom*