ECE 477 Final Report – Fall 2010
Team 3 – OMNiMOTE

Team Members:

#1: __Matthew Barga _____________ Signature: ______________________ Date: _12-13-10_

#2: __Nathan Meyers______________ Signature: ______________________ Date: _12-13-10_

#3: __Alex Reyes _________________ Signature: ______________________ Date: _12-13-10_

#4: __Joe Riley___________________ Signature: ______________________ Date: _12-13-10_

CRITERION	SCORE	MPY	PTS
Technical content	0 1 2 3 4 5 6 7 8 9 10	3	
Design documentation	0 1 2 3 4 5 6 7 8 9 10	3	
Technical writing style	0 1 2 3 4 5 6 7 8 9 10	2	
Contributions	0 1 2 3 4 5 6 7 8 9 10	1	
Editing	0 1 2 3 4 5 6 7 8 9 10	1	
		TOTAL	

Comments:

TABLE OF CONTENTS

Abstract

The OMNiMOTE is a Wi-Fi based Universal Remote Control. It can connect to your wireless network, or create an ad-hoc network of its own, and allows you to control all of your IR components through any supported Wi-Fi device. It can learn IR commands from any remote using a supported protocol, enabling you to add any button from your existing remote. Once learned, the OMNiMOTE can send that command to the device through an infrared LED that will be attached to the intended component.

1 Project Overview and Block Diagram

Figure 1.1: Photo of OMNiMOTE

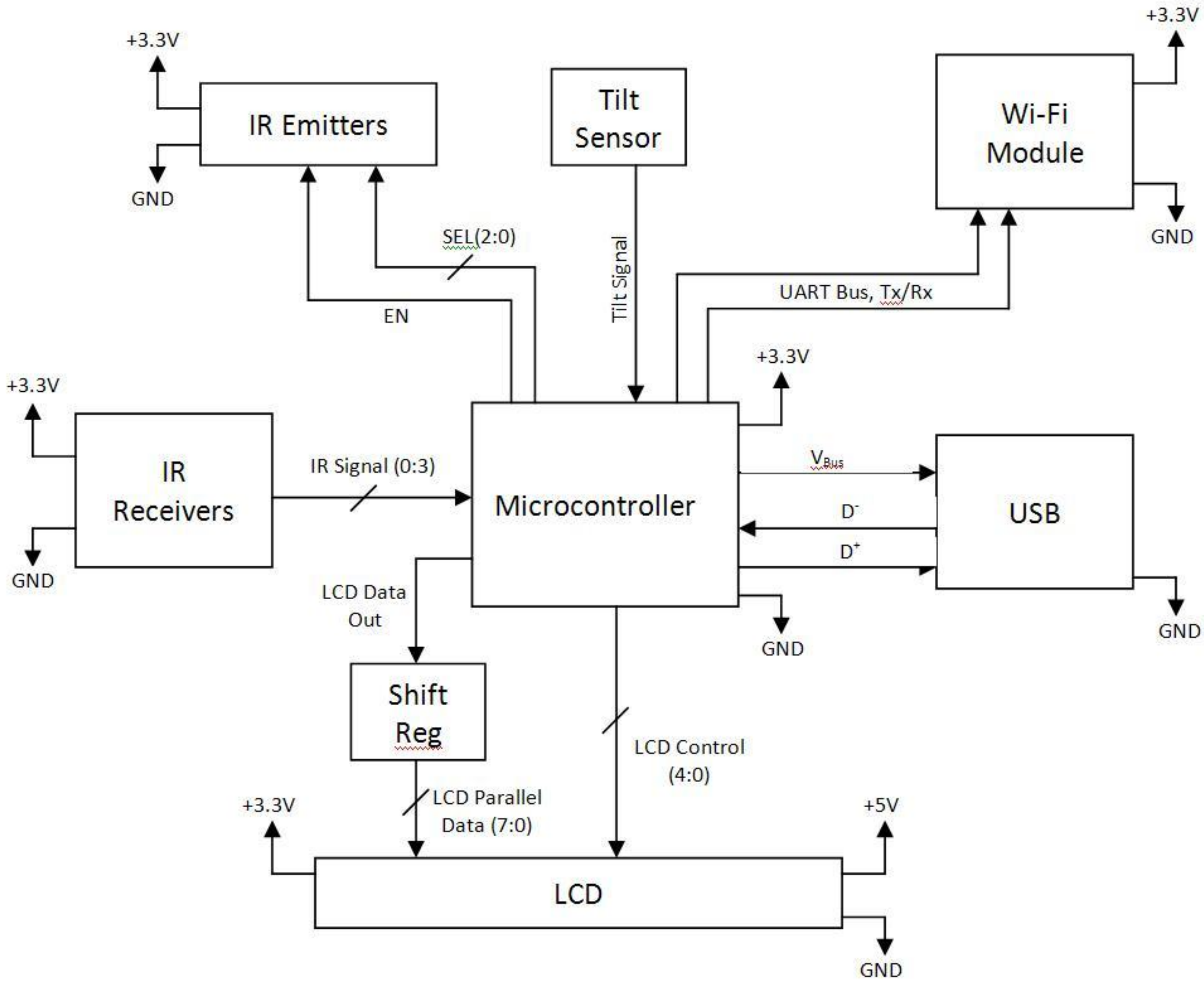

Figure 1.2: Block Diagram of OMNiMOTE Circuitry

2 Team Success Criteria and Fulfillment

2.1 An ability to transmit data over unsecured, WEP, WPA, and WPA2 networks.
2.2 An ability to upload wireless network settings via USB.
2.3 An ability to detect orientation to display information on LCD accordingly.
2.4 An ability to decode and store IR commands from a variety of protocols and frequencies.
2.5 An ability to encode and transmit IR commands in a variety of protocols and frequencies.

Currently, all 5 PSSCs have been fulfilled.

3 Constraint Analysis and Component Selection

3.1 Introduction

The OMNiMOTE is a device that is doing the same work, just better, as a device that consumes 2 AAA batteries for over a year. Power consumption, IO support, and high clock speeds should be optimal in our ideal components. We want our components as small and reliable as possible, while also making them cost effective. The OMNiMOTE must be capable of delivering an excellent experience to the user, while also not breaking their wallet or power bill.

3.2 Design Constraint Analysis

The OMNiMOTE works as a tool to change how a user interacts with his home theatre equipment. It will be always plugged in and waiting for commands from its Wi-Fi connection, which means we will want to consider power efficiency as an important design constraint. It must be fast enough to emulate the experience of IR remotes in terms of response time between button presses and emitting IR commands, otherwise users may get frustrated. Emitting IR commands requires our microcontroller to be able to send modulated signals at frequencies of 36kHz, 38kHz, 40kHz, and 56kHz in order to output the most common IR protocols.

3.3 Computation Requirements

To be a successful product, the OMNiMOTE must be able to perform fast enough that the user does not get frustrated with the lag between pressing a button on the interface and the command being sent to the component. Focus group results have shown this maximum time should be is no longer than 100ms. The microcontroller must also provide a PWM with the ability to modulate at 56kHz, 40kHz, 38kHz, and 36kHz. This is important for emitting the IR commands at their correct carrier frequencies. Because we are using USB, we need the microcontroller to be able to handle the data speeds of USB, so it require a clock speed much higher than 12MHz (the speed of USB1.1). While controlling all of this, it will also have to be able to update an LCD with 32 x 122 pixels.

3.4 Interface Requirements

The OMNiMOTE has 5 external devices, plus the USB, that the microcontroller must use. The external components are the IR receiver circuit, LCD, IR emitter circuit, tilt sensor, and WiFi module. The USB, integrated into the controller, will have to use a specific set of IO pins. I would suggest at least 48 IO pins, giving the IR circuits 10 total, the LCD at least 7, the WiFi module 6, and the USB 4. The tilt sensor would only need one, and the remaining 20 for backup and debugging.

3.5 On-Chip Peripheral Requirements

The OMNiMOTE will be able to output IR commands with up to 8 IR blasters. The microcontroller in the OMNiMOTE must have a PWM that is capable of modulating at 56kHz, 40kHz, 38kHz, and 36kHz. It must also have at least one UART connection to communicate with the Wi-Fi module, and have at least 2 high speed IOs for USB traffic. The microcontroller must be capable of at least 2 interrupts, and 2 timers for updating the LCD, and controlling the IR receiver code. For storing data from IR commands and the interface, a large amount of flash memory will be needed to store enough button information to be useful. For reading in the IR commands, we have to have enough ram to record the changes in data pulses at each of the four frequencies, and then decode the right one. Our code required 280 integers, which would be 1120 bytes, just for the IR receivers. The OMNiMOTE can have no less than 10K of RAM.

3.6 Off-Chip Peripheral Requirements

The OMNiMOTE will require a Wi-Fi module capable of handling at least consumer 802.11b connections, though if we would want to have a faster connection, using 802.11g would be useful also. An LCD will give the user feedback on internal settings, as well as instruct them on set-up. An tilt sensor is used to relay the OMNiMOTE's current orientation to ground to display information on the LCD accordingly.

3.7 Power Constraints

Because the device is meant to be used in a home theatre setup, it will be powered on AC power only. The LCD display module requires the largest voltage, 6.5 volts, which determines our DC requirement. The OMNiMOTE will be replacing remote controls that normally operate with very low power consumptions, so it should use as little power as possible to function. We expect the device to use a 500mA power supply, but we estimate the device will only use around 410mA at max. The OMNiMOTE would, under normal conditions, be placed in a well-ventilated area, but if this is not the case, the packaging will have sufficient ventilation to keep the internals at their recommended operating temperatures. The temperature range sufficient for all components to successfully operate is -10°C - 70°C. With ventilation, room temperature operation (around 25°C), any sized packaging will be sufficient for normal conditions.

3.8 Packaging Constraints

Packaging choices are very critical for a device that will be part of a customer's home theatre. We know our users will not have a large amount of space, and to store the device, they are very likely to have shelving to store them. Shelving heights tend to be around 6 inches tall or higher, so the height of the OMNiMOTE, while standing vertical, should be no taller than 6 inches. The width of the device should primarily be determined by the components inside, as well as the depth. The average height of consumer home theater equipment is 6" tall, so this will be the maximum height. Those devices tend to be around 12" deep, so any depth smaller than that is fine. We know we don't want the OMNiMOTE to be too wide in its vertical orientation, so making it no wider than 3" should be sufficient. Making the packaging as small as possible, as well as a great design, will attract our potential customers from other choices and increase its profitability. The design of the packaging must also cater to ventilation to keep the internal components in the recommended operating temperature.

3.9 Cost Constraints

For the OMNiMOTE to sell well, it needs to be priced comparably to other universal remote control solutions. Universal remote controls with the ability to learn IR commands, like the RCA RCU900, can cost as little as $50. However, universal remotes that eliminate line-of-sight like the OMNiMOTE can start at over $100. A device found supporting the most similar features to the OMNiMOTE is RedEye by ThinkFlood©, with an MSRP of $188. This price will be the maximum set for our cost constraints.

3.10 Component Selection Rationale

The microcontroller is the most important part of the OMNiMOTE, and we have specific requirements for it. The LPC2377 and the PIC32MX340F were our two choices for microcontrollers. These microcontrollers have many of the same features, the biggest difference, however, is their architecture and supported libraries. One of the most important design criteria is power management, and the LPC2377 is able to run at 48MHz typically around 42mA, where the PIC32MX340F consumes 55mA typically, with a maximum of 80mA.   The most important calculation requirement we having enough ram. Both of these processors have 32K of ram, which is plenty, and they are both capable of running at a fast enough frequency to ensure fast response times from user input.

The LPC2377 has more than enough IO for our needs, has the extensive ARM architecture, is the most power efficient chip for all of its capabilities, and a development kit is available for free. Wi-Fi is the most important peripheral on the OMNiMOTE, and the Roving Networks WiFly GSX and the Lantronix WiPort were our two choices. Both chips are similar, featuring on-board TCP/IP stack management, support of the faster 802.11G protocol, and automatic data packetizing. While the WiPort is much more robust in its features, it cannot handle WPA2 encryption like the WiFly can, and is more expensive. The WiFly was chosen as our Wi-Fi module for these reasons.

3.11 Summary

When designing the OMNiMOTE, we need to have the user experience as our top priority. Power draw is a very important issue, and we chose our parts accordingly. Cost is also a large constraint, so many of the components are chosen because of their cost efficiency. We wanted to make sure that all of our components were able to have more functions (IO pins, timers, settings) than we require now, but not too many to be over necessary.

4 Patent Liability Analysis

4.1 Introduction

The OMNiMOTE is a Universal Control Station for up to eight IR controlled devices. The OMNiMOTE will host a virtual user interface which WIFI enabled devices will be able to obtain via the OMNiMOTE's WIFI module. This interface will provide the user with options to record IR commands using the IR receivers on the base station or issue previously recorded IR commands, to one of the eight IR emitters connected to the base station. Some of the functions under parent liability are detecting and recording IR coded signals, storing user interface for virtual remote control, matching user interface action with an IR coded signal and emitting previously recorded IR coded signals.

4.2 Results of Patent and Product Search

A patent search for the main functions of the OMNiMOTE resulted in three potential liabilities for the OMNiMOTE design. These three results describe functions similar to those used in the OMNiMOTE design.

The first is Patent No. 5 602 664 filed February 11, 1997 and describes an IR repeater. The main sections of the device are a receiver section, an oscillator section and an emitter section. The receiver section detects an IR coded signal. The oscillator generates a modulating signal used by the transmitter section. This last section is coupled to the receiver section and a modulating oscillator and generates an IR light signal matching the detected coded signal by the receiving section. [4.1] One of the key claims of this patent is presented below.

"1. An IR repeater comprising:

a receiver section, for receiving an IR light signal representing a coded signal modulated by a modulating signal;

a first IR detector coupled to said receiver section for detecting said IR light signal and generating a signal representing said coded signal;

a fixed-frequency oscillator, for generating a transmitter modulating signal, said fixed-frequency oscillator generating said transmitter modulating signal at a predetermined frequency independent of the frequency of the received modulating signal; and

a transmitter section, for generating an output IR light signal representing said detected coded signal modulated by said transmitter modulating signal." [4.1]

The second is Patent Application No. US 2006/0020999 published January 3, 2006 and presents both systems and methods for connecting infrared controllable devices to digital networks. The methods comprise digital-signal-receive logic and convert logic to convert the received signal to an IR command signal for an IR-controllable device. [4.2] Some of the key claims of this application are presented below.

"3. A system comprising:

a digital input port configured to receive a digital signal from a digital network;

a converter configured to convert the digital signal into an infrared (IR) command signal, the IR command signal being configured to command an IR-controllable device; and

an IR transmitter configured to transmit the IR command signal to the IR-controllable device.

4. The system of claim 3, further comprising:

an IR code library having an IR code, the IR code corresponding to an IR command signal for an IR controllable device.

5. The system of claim 4, wherein the converter is further configured to access the IR code library, the IR code library being accessed for an IR code that corresponds to the received digital signal, the converter further being configured to receive the IR code from the IR code *library, the converter further being configured to generate the IR command signal from the received IR code."* [4.2]

The last is Patent Application No. US 2010/0134338 A1 published June 3, 2010 and presents a method providing a virtual universal remote control interface that detects user interaction which is ultimately mapped to an IR coded device function. [4.3] The key claims from this application are presented below.

"1. A computer readable storage medium including computer executable instructions for providing a virtual, universal remote control feature, the instructions comprising instructions to:

display a virtual remote interface for an electronic device;

detect a user interaction with an element of the virtual remote interface;

map the virtual remote interface element to a device-function code; and

provide the device-function code to a code transmitter, wherein the device-function code corresponds to a control command associated with the electronic device.

6. The computer readable storage media of claim 1, wherein the virtual remote interface element is one of a plurality of virtual remote interface elements wherein the instructions to display include instructions to access a device profile including information indicative of positioning and label of the virtual remote interface elements."[4.3]

4.3 Analysis of Patent Liability

Patent 664 is perhaps the potential liability with most differences from the OMNiMOTE design. Therefore, any potential infringement would be under the doctrine of equivalents. Both designs are composed of substantially the same sections and perform the substantially the same function. Both devices contain a receiver section, a section which generates a carrier frequency, and an emitter section.

However, key differences exist between the designs. While one of the patents claims describes an oscillator section, which has a fixed frequency, the OMNiMOTE uses a PWM to generate four different carrier frequencies.

Furthermore, the overall process of the two designs are differentiated by the fact that during the OMNiMOTE’s normal operating sequence, an IR coded signal will not travel directly from the transmitter section to the emitter section, otherwise being stored in memory to be generated at a later time. The transmitter section of the OMNiMOTE is not coupled to the receiver section as the method in Claim 1describes.

The next greatest potential liability resulting from the patent search is Patent Application No. US 2006/0020999. The claims in this patent describe a system comprising of a converter and a transmitter section similar to the OMNiMOTE design. However, claims 4 and 5 of Patent Application US 2006/0020999 describe an IR code library and a converter able to access said library. The OMNiMOTE’s design saves IR codes which a microcontroller is able to access to match and decode device control instructions. This aspect of the design might constitute literal infringement.

The last patent search result poses the greatest risk for patent liability. Patent Application No. US 2010/0134338 A1 presents a method very similar to that used in the OMNiMOTE’s design. Both methods store information about a virtual control remote on a computer readable format. In addition, claim 6 of the patent describes a method identical to that of the OMNiMOTE’s control configuration process.

One of the differences between the two methods is that the OMNiMOTE provides only a description of the virtual interface and not actual executable instructions as described in claim 1 of the patent. The detection of user interaction and display of the virtual remote control would be handled by a dedicated application that is not part of the OMNiMOTE design.

4.4 Action Recommended

In the case of US Patent No. 5 602 664 declaratory judgment action is recommended given that there are clear differences between the patent and OMNiMOTE design, namely, signal frequency and operating sequence.

In the case of Patent Application No. US 2006/0020999, if it were to result in a patent, declaratory judgment action is recommended with a focus on claim1 where the patent states that the system must be coupled to a digital signal network. The literal infringement on claims 4 and 5 might be avoided if the OMNiMOTE system is differentiated from the system described in claim 1. If it were to still hold as a subsystem infringement we might have to obtain a license for manufacturing.

In the case of Patent Application No. US 2010/0134338 A1, if it were to result in a patent, declaratory judgment action is recommended for infringement related to claim 1 since there are differences in what information is stored in the storage device. For infringement on the basis of claim 6, it should be possible to design around the configuration of the remote by making a simple interface where the arrangement of the components depends only on the number of elements. This way information about the positioning of the elements would not need to be stored.

4.5 Summary

Three results from a patent search pose potential liabilities for the OMNiMOTE design, specifically relating to functions for recording and emitting IR coded signals, storing user interface for virtual remote control, matching user interface action with an IR coded signal and emitting previously recorded IR coded signals. These three results are US Patent No. 5 602 664, Patent Application No. US 2006/0020999 and Patent Application No. US 2010/0134338 A1 presenting an IR repeater, a method to connect infrared controllable devices to digital networks, and a virtual universal remote control respectively. Due to some clear differences between the designs declaratory judgment action on all claims by the first two results and on some cases of Patent Application No. US 2010/0134338 A1. Some claims of this last application might require modifying the OMNiMOTE and designing around the method described in the application.

5 Reliability and Safety Analysis

5.1 Introduction

The OMNiMOTE is a universal remote control, designed to allow the user to control their infrared components with any web-enabled device. Since it will ideally be used to replace all control to many devices, it needs to be very reliable. If the device fails, you could potentially lose control of your TV, VCR, DVD player, sound system, and any other components that you may be controlling via a remote.

The only real safety concern with the OMNiMOTE is heat dissipation. The device will rarely be unplugged or turned off, meaning that most of the parts will have a continuous current flow for a long period of time. This current flow will likely cause components to heat up. If any parts get too hot, it could damage surrounding objects or even start a fire. The OMNiMOTE tries to compensate for this risk by running all components well within their recommended operating conditions.

5.2 Reliability Analysis

The components most likely to fail in the OMNiMOTE design are the microcontroller, due to its complexity, the power regulators, due to their high operating temperatures, and the LCD display, due to its complexity. Tables 5.2.1, 5.2.3, and 5.2.4 show all parameters required to calculate the MTTF for each component. More information regarding each choice is explained after each table.

The Wi-Fi module is also a very complex component that will probably have a high failure rate. However, after speaking with the manufacturer and looking at countless datasheets, the required information used to determine the failure rate estimates could not be located. The datasheet does show that the module uses a 32-bit processor, but this alone is not enough to come to a meaningful estimation of its failure rate. If this were an actual product being marketed, we would need to run a test on the specific module we are using to determine its failure rate.

To calculate the failure rates of each component, many assumptions were made. The parameters π_E, π_Q, and π_L were constant throughout each calculation. π_E was chosen because the device will be in a stationary environment. π_Q was chosen because the screening levels for each component were unknown. π_L was chosen because each component we analyzed has been in production for more than two years.

Table 5.2.1: Microcontroller

Parameter name	**Description**	**Value**	**Comments**
C_1	Die complexity	0.56	32 bits, MOS logic
C_2	Package Failure Rate	0.077	Interpolated from Table 5.9 in [5.1]. 144 pins, non-hermetic.
π_T	Temperature Factor	0.26	Interpolated from Table 5.8 in [5.1]. T_J calculation shown in Table 5.2.2.
π_E	Environment Factor	2.0	Ground, stationary
π_Q	Quality Factors	10	Other commercial or unknown screening levels
π_L	Learning Factor	1.0	> 2 years in production
λ_P	Failure Rate/10^6 Hours	3.0	Failures per million hours
MTTF	Mean Time To Failure	3.3×10^5 Hours 38 years	$(1/\lambda_P)*10^6$ Hours

Table 5.2.2: Microcontroller T_J Calculation

Parameter name	**Description**	**Value**	**Comments**
T_C	Case Temperature	45	Ground, stationary
θ_{JC}	Junction-to-case thermal resistance	10	Die area > 14,400 mil^2. Die area found in [5.4], page 3.
P	Maximum power dissipation	0.165	3.3V * 50mA = 0.165W.
T_J	Junction Temperature	46.65	Average Junction Temperature

The die complexity and package failure rate were standard values taken from the MIL Handbook [5.1]. The temperature factor was a little more complicated because we needed to calculate the junction temperature. To calculate the junction temperature, we used the equation given in Section 5.11 of the MIL Handbook [5.1]. For the power dissipation, we connected our PCB with just the microprocessor being powered and measured the consumed current to be 35 mA. We then rounded that measurement up to 50 mA to account for any errors.

Table 5.2.3: Power Regulator

Parameter name	**Description**	**Value**	**Comments**
C_1	Die complexity	0.01	Linear, 1-100 transistors
C_2	Package Failure Rate	0.0034	8 pins, non-hermetic DIP
π_T	Temperature Factor	58	T_J (125C) taken from [5.3], page 2
π_E	Environment Factor	2.0	Ground, stationary
π_Q	Quality Factors	10	Other commercial or unknown screening levels
π_L	Learning Factor	1.0	> 2 years in production
λ_P	Failure Rate/10^6 Hours	5.9	Failures per million hours
MTTF	Mean Time To Failure	1.7×10^5 Hours 19 years	$(1/\lambda_P)*10^6$ Hours

The power regulators didn't require a whole lot of extra calculation. The die complexity was estimated from the regulator being a linear device with less than 100 transistors. The package failure rate was determined from the regulator being in an 8-pin, non-hermetic DIP package. To calculate the temperature factor, we assumed the junction temperature to be the absolute maximum.

Table 5.2.4: LCD Display Controller

Parameter name	**Description**	**Value**	**Comments**
C_1	Die complexity	0.14	8-bit MOS
C_2	Package Failure Rate	0.052	Interpolated from Table 5.9 in [5.1]. 100 pins, non-hermetic.
π_T	Temperature Factor	0.30	Interpolated from Table 5.8 in [5.1]. T_J calculation shown in Table 5.2.5.
π_E	Environment Factor	2.0	Ground, stationary
π_Q	Quality Factors	10	Other commercial or unknown screening levels
π_L	Learning Factor	1.0	> 2 years in production
λ_P	Failure Rate/10^6 Hours	1.46	Failures per million hours
MTTF	Mean Time To Failure	$6.9\text{x}10^5$ Hours 78 Years	$(1/\lambda_P)*10^6$ Hours

Table 5.2.5: LCD T_J Calculation

Parameter name	**Description**	**Value**	**Comments**
T_C	Case Temperature	45	Ground, stationary
θ_{JC}	Junction-to-case thermal resistance	22	Die area < 14,400 mil^2. Chip area found in [5.6], page 8, to be 19,353 mil^2. Die area is assumed to be much less.
P	Maximum power dissipation	0.25	0.25W, taken from [5.6], page 35.
T_J	Junction Temperature	50.5	Average Junction Temperature

A lot of assumptions were made when calculating the failure rate for the LCD display. After looking at the datasheet, there was not enough information to calculate the MTTF of the display itself. Instead, the failure rate of the LCD controller has been calculated. The LCD uses an SBN1661G_M02 controller, as stated in its datasheet [5.5]. The controller data sheet shows that it is a 100-pin chip with an 8-bit microcontroller [5.6]. These were used to determine the package failure rate and die complexity. To determine the temperature factor, the thermal resistance and power dissipation needed to be found. The thermal resistance was found by assuming the die size of the controller to be less than 14,400 mil^2. The controller datasheet specifies that the entire chip has an area of 19,353 mil^2, so it is assumed that the die size will be less than this. The power dissipation was taken from the maximum value specified in the datasheet, since we have no way of determining the controller's sole power dissipation.

There are several ways to increase the lifespan of our parts. One such way is to heat sink the components. The regulators in particular would benefit from heat sinking, since the maximum junction temperature was used in its calculation. Another possible way is just to use higher quality parts. In each component, the quality factor was assumed to be 10 since we didn't actually know the screen process. If we purchased higher quality products, we could reduce this by a factor of 5 or 10.

5.3 Failure Mode, Effects, and Criticality Analysis (FMECA)

Our schematic can be divided into 7 different subsystems: microcontroller, LCD display, IR emitters, IR receivers, Wi-Fi module, power supply, and USB interface. The subsystems that will most likely fail first are the power supply, microcontroller, and LCD display. The failures, causes, and symptoms for all subsystems can be seen in the FEMCA worksheets in Appendix B.

There will be three criticality levels associated with the OMNiMOTE. A high criticality failure is a failure that could possibly cause harm to the user. This only refers to a short circuit or power supply failure in which the product could potentially catch fire. Currently, there are no precautions in place for this problem, but one possible fix is to include a fuse. That way, if the product causes a short circuit and draws too much current, the fuse will blow and the problem will be under control. All subsystems can cause a high failure if they cause a short circuit.

A medium criticality failure is a failure that causes the device to be non-functional in almost all aspects that it was meant to function. For instance, since the device is designed to control your IR components, if the IR emitters fail then you can't really do anything that the device was meant to do. The possible medium failures include IR emitters, IR receivers, and the Wi-Fi module.

A low criticality failure is a failure that causes some part of the device to not function, but without causing the device to be useless. For instance, if the LCD display stops working, the device can still function but the user will be required to read all status information from the user interface. The device can still control your components, but at a little less convenience than it was intended. The possible subsystems with this type of failure are the USB interface and LCD display.

5.4 Summary

Overall, the OMNiMOTE is very reliable. The component with lowest lifespan is the regulators at around 19 years to failure. As far as electronics go, this is pretty good, but there is still room for improvement. For instance, if we added heatsinks to the voltage regulators, their lifetime would be extended even further than 19 years. This would increase the life of the OMNiMOTE also, assuming that the regulators were, in fact, the most likely component to fail.

6 Ethical and Environmental Impact Analysis

6.1 Introduction

The OMNiMOTE is designed to be another component in the user's home theater system. It will be used indoors and will be compact enough to fit onto a shelf. The design also allows for vertical or horizontal orientation. The circuit construction utilizes 4 PCBs, the reason being to allow for flexibility in the mounting of jacks for external component access inside of the case. The primary components on the main board are the microcontroller and the wifi module. The other three boards separately contain IR receivers, mono jacks, USB input connectors and the LCD screen and associated circuitry. Communication with the device will be done over wifi, USB and IR, however all controls and settings that are established will be done over the USB and wifi interfaces.

There will be some ethics issues to address in securing the data that will be transferred over the USB and wifi connections, as well as ensuring correct data transmission to prevent any erroneous behavior of the product. As the product will mostly be used inside the home, there will also need to be tests for possible failure modes if the device is dropped, if it has food/drink spilled on it, etc. As households may have children or animals present, testing should be done to find ways that they could possibly become injured, either from the device falling or from possible electrical shock. From an environmental standpoint, the disposal of the casing will be the biggest problem when the device reaches the end of its life cycle. Responsible research into component selection and PCB printing should be done to reduce the amount of lead and other harmful materials contained in the device.

6.2 Ethical Impact Analysis

The first major ethical challenge that will be addressed is the correct transfer of data over the wifi. The user interaction with the device will be primarily done over a wifi connection. After the user creates a new button in a defined interface on an external wifi enabled device, if he or she presses the button, the button action will be sent via wifi to the OMNiMOTE station and the appropriate action will be taken. There is risk of incorrect data being received by the OMNiMOTE station. If the wifi module is to be put to sleep (to save power), it can be set to wake up when it detects data being sent on the UART line. The consequence of this is that the first 2 or 3 bytes received will be lost before the device wakes up. Also, as with any wireless communication protocols, there is potential for external interference, either by other wifi signals or any signal within the 802.11x frequency range. It is worth mentioning here that the wifi module employed is FCC compliant [6.2]. To prevent the loss of important data at the beginning of a transmission, signals could be preceded with a repetition of a particular byte or message (something not relevant in a regular data transmission), and when a break from this sequence is detected, the data can be parsed from that point. As the user's home could likely be in range of other wifi networks, redundancy in the message data sent over the wifi network could help overcome the problem of dropped or incorrectly received bytes within the data message. This could be done by having the application sending data to the wifi repeat every byte within a message a number of times and having the input routines on the microcontroller take this into account when parsing messages by checking for a repetition of a byte before actually considering that byte as valid data. The other option is to do error checking of any protocol message received and to send a retransmit request to the controller application if an unexpected message is received.

The consequences of incorrect data being sent to the wifi include unintended actions taken by the device. The possibilities include incorrect signals being sent out over the IR blasters, in which case no action or a wrong action might be taken by the IR controlled device. The worst case scenario in this case would be a waste of the user's utilities if a number of media components are turned on unbeknown to the user, or if a signal is continuously output on an IR output line. Unintended functional behavior of the wifi module is another possibility. All wifi settings will be established through communication with the microcontroller over UART. If the wrong configuration settings are given, the wifi module could get put into a mode where it is never put to sleep. This situation could also result even if the wifi module is still in 'wake on UART' mode if for some reason there is a constant stream of meaningless data on the UART line. The only way to prevent this is to restrict the settings that the user is allowed to give to the wifi, and test for any case where unintended data could be streaming on the UART line. It is also possible to query the wifi module from the microcontroller to read running statistics and confirm if it is currently set to go into sleep mode when idle.

There is also the remote possibility of someone hacking the wifi network. There is no sensitive data that will be sent over the network, only IR control signals. The only concern that this would bring up would be that anyone who manages to connect to the network could potentially gain control of the OMNiMOTE unit. All of the IR signals that are collected will be stored either in the flash or RAM memory of the microcontroller. This means that these signals are potentially accessible to anyone on the network and can be sent out over the IR blasters. There is a message protocol that has been developed by our team for communication between the controller application and the base unit (the wifi module just transparently forwards bytes; there is no protocol to communicate with it aside from entering it into command mode for configuration) however, so the unintended user would have to be familiar with this protocol to be able to efficiently access any of the IR control signals and take any of the actions associated with them. The wifi module also makes use of WEP or WPA/2 encryption to deter any undesired accesses to the network. Even if someone was able to access the network and had knowledge of the communication protocol, the most damage that could be done would be similar to the consequences mentioned in the previous paragraph, namely, a waste of utilities if a number of media components were turned on and left on for long periods of time or signals were continuously output. A last point to mention is that there is a reset button to return the wifi module to factory settings, so if it somehow took on a different SSID or the user forgot the WPA password, it could be recovered to a known state.

The possibility of IR interference is also an issue. It is possible that due to a bug, an IR command could be continuously sent out over an IR blaster. This could be a waste of power and a nuisance if a particular command was being sent out (e.g. TV being turned on and off in rapid succession or IR LED constantly on). It could also interfere with operation of other IR remote devices if signals got crossed. If an IR output was stuck on it should not interfere with operation of other IR devices however, as the control of components is based on the timing of rising and falling edges in the IR signals.

There is also concern of physical damage to the product. In a user's household, there could be children or animals that could easily spill food on the device or cause the device to fall, resulting in damage possibly beyond repair. There are no moving parts inside the device, so physical shock should not cause any extensive damage to the circuitry. However, if the case became dented or broken, this could cause damage to the internal components. There should at least be an error message displayed on the LCD if the device gets into a 'soft' failure mode and is unable to be operated. There is also a heartbeat signal on the board that flashes a green LED in normal operation. If this light is stuck on or off, then there is an error. Responsible action should also be taken in regards to case design to ensure that it is robust enough to at least handle a minor fall, and that the internal components are isolated from the outside to prevent any foreign material from getting in the case and causing a problem.

The most concern for potential harm to the user would be a power circuit failure. This could cause fires or shock to the user if handled improperly. The highest potential for this kind of failure would be in the event of a short circuit. This could occur if traces were bridged if liquid is spilled into the device or some other conductive material somehow came in contact with the PCB. To prevent any foreign material from getting into the device, the casing should provide adequate protection for the internal components. Short circuits could essentially be turned into a minor issue if proper fuses are used. Lastly, proper ventilation for the device should allow for any excess amount of heat to be dispersed so that the functionality of the device is not jeopardized. If for some reason the device ends up in a confined space, there could be heat buildup which could affect operation and life-span. There will be ventilation holes on the casing to help disperse any head generated by the device.

6.3 Environmental Impact Analysis

There are a number of environmental impact issues to address over the life of the device as well. The main environmental concern involved with the production of the device is the responsible disposal of the casing. The OMNiMOTE will be encased in aluminum with acrylic windows on the front and back. This will contribute to the aesthetics of the device and the robustness of the device. The aluminum will be recyclable, but it will be up to the user to do this. It should be easily removable from the rest of the casing, and a label should be placed on the inside to remind the user to recycle it, improving the chances that the user will actually take the actions to recycle it.

As the device does have PCBs, there will be small amounts of lead used in the traces. Exposed lead is toxic to the environment and the proper actions should be taken to address this problem in the OMNiMOTE design. To be responsible, many PCB printing companies should be researched to find one that uses minimal lead and other environmentally unfriendly materials in their manufacturing process. The consequence of this is the price increase due to the higher expense of more 'green' practices.

Use of components in compliance with RoHS standards is also a way to reduce environmental impact. Through choosing components that are in compliance with RoHS standards, and responsibly manufacturing the PCB, the amount of harmful materials used in the device and in its manufacturing process can be reduced [6.1]. The wifi module is FCC and RoHS compliant [6.2], so not only does it contain minimal amounts of lead, it is also compliant with communication regulations. The majority of other components selected are also RoHS compliant. It is also essential to give the user instructions on how to responsibly recycle the PCB as it contains materials that could be harmful if left in a landfill. The EPA and other organizations will take used electronics for disposal [6.3]. Instructions and links for finding agencies that will dispose of the PCB [3] should be placed in the user manual, and a sticker should be placed on the bottom of the actual product reminding the user to look at the manual for disposal instructions.

Finally there is the issue of power consumption. It is worth noting that the OMNiMOTE will consume very little power during normal functionality. The microcontroller and wifi will only need to be awake when IR signals are being read or sent out. Besides them, the only component that will draw a reasonable amount of power will be the LCD screen, which is always on. If it was too much of an issue, the LCD could be turned off after a period on inactivity. The only essential information it displays is wifi connection strength and current status messages. The entire main board including the LCD could maximally consume ~500 mA of current at a 9V input DC power supply depending on whether or not the microcontroller and wifi are asleep and whether the IR blasters are sending out signals. For the majority of normal use, the device has been tested to operate consistently at around 60-70mA, resulting in approximately 630mW of power consumption.

6.4 Summary

A detailed summary of potential ethical and environmental impacts have been outlined above. The highest concern regarding ethical impact involves the correct transfer of functional control data that is sent over wifi. The use of various kinds of error checking can ensure that no unintended actions are taken as a result of an incorrectly received data signal. Proper testing should also be done to eliminate or reduce the severity of possible injury to children or small animals who may be in a household. Most of the solutions for this problem can be reached by looking into various case designs. Responsible actions should also be taken to mitigate any possible impacts on environment that the device may have. This includes taking time to research components and PCB manufacturing techniques that use minimal amounts of chemicals and materials that are harmful to the environment, as well as educating the user on the proper way to recycle components of the device at the end of its life cycle.

7 Packaging Design Considerations

7.1 Introduction

The OMNIMOTE will be 8.25” x 2” x 6”, and will sport a luxurious aluminum casing. The front, bottom, and back panels are made of a dark-tinted acrylic, which allow the IR receiver’s at the top to receive data and be protected inside the casing. All of the mono-jacks for the IR blasters, the USB connector, power switch, and power connector are mounted on the back panel. This keeps all of the clutter from the long IR blaster cables tucked away, while everything the user will interface with most of the time is on the front. This includes the LCD and the IR receivers for learning commands.

7.2 Commercial Product Packaging

Two products that are available to consumers today are the Bitwise Controls Remote Edition and the RedEye. Each product comes close to performing what the OMNIMOTE can, but each product seems to neglect its user, or miss a key feature. While the Bitwise Controls Remote is also able to provide a vast amount of other home automation features, the casing is “all function, no fashion”, and the location of some of the ports don’t seem to be really well thought out. The RedEye seems to have thought of the user, its packaging contains a bright LED, and with its line of site requirement, cannot be hidden away.

7.3 Product #1

The Bitwise Controls Remote Edition, the main product actually referred to the BC4X1 Pro Automation Controller, offers a lot of functionality in a very small package, but it’s clear that the design of the packaging uses dull materials and looks like most enclosures available online, with the back featuring most of the inputs. This design would seem useful, but the front also has 2 inputs, an Ethernet and AC adaptor input. Also featured on the rear is the IR receiver, for the device to learn commands. This seems like a mistake that could potentially aggravate users, something the OMNIMOTE packaging will not have. The packaging for the BC4X1 is small, capable of fitting into any home theater, which is also what the OMNIMOTE will achieve.

7.4 Product #2

The RedEye device seems to be marketed more for home use than commercial use with Bitwise Controls, and its packaging shows that. Being made to look like an iPod dock, the RedEye features an iPod charging dock on top, as well as an IR receiver in the front for learning commands, and an IR emitter in the back to transmit commands to the components. The packaging of the device is very small, smaller than the OMNiMOTE, and very easy to implement into a consumer’s home theater. An issue the casing has is the choice of having very bright LEDs on the front, and it only has one emitter, so it must be with-in all of the components line-of-sight. Without these blasters, however, the RedEye is able to achieve a sleek, modern look, which is a style the OMNIMOTE will achieve, however instead of the cheap plastic look, it will have the luxurious look of aluminum.

7.5 Project Packaging Specifications

The design of the OMNIMOTE stems from designs of luxury devices from companies like Apple. The main body of the OMNIMOTE is made of aluminum, and stands 6” tall, 8” deep, and 2” wide in the vertical orientation. This size requirement is so the device will be able to fit in almost every consumer entertainment center. The other part of the packaging will be made of acrylic, and to an aluminum tray, covered in rubber for protection from shorting, which holds the PCB onto the bottom panel. This allows the LCD to be seen through it, as well as the IR inputs to read commands.

The top aluminum section of the housing is connected to the acrylic by 12 screws, 6 which are connected to the bottom acrylic, and the other 6 attached to inside aluminum beams that are connected to the front and back panels. This makes the device have a clean presentation, but also makes assembly/disassembly easier for repairs. The holes are tapped and threaded to lock the casing and the PCB tray to the acrylic. The IR receivers are at the top, in the front, to provide the user easy access when programming a new command into the interface, as well as feedback from the LCD right below. All external cables are pushed to the back of the device, to allow hiding them easier, making a cleaner experience than some competitors. The back panel is milled with holes that are to spec with all of the external connectors.

7.6 PCB Footprint Layout

The components that have been selected for the OMNIMOTE are very small, and fit on a small PCB within the device. To The largest parts of the design the power circuit, the microcontroller, the IR circuit, and the Wi-Fi module. We are allowing the power supply to have at least a 2” x 1” footprint on the PCB, the IR circuit will have at least a 1.5” x 1” footprint, and the Wi-Fi module will have at least a 1” x 1” footprint. The IR circuitry and the USB connector are on a separate PCB, and are attached to the back panel. As shown in Appendix C, the 5”x 5” PCB should allow sufficient room for all mounted hardware, while still giving enough room for reliable wire harnesses. The LCD and IR input receivers will not be mounted on the board, but connectors will be mounted at the top and middle of the front of the PCB.

7.7 Summary

While there are not many choices of similar products to the OMNIMOTE, most do not offer the design styles of most high-end components. By putting the user as the primary focus of the design, the OMNIMOTE is able to provide the best overall user experience. The sleek aluminum look will compliment high-end stereo equipment, and will look good in any setting. The modular design of the IR blasters allows users to only have the cables they need, and all of the connections being in the back allow for easy cable management.

8 Schematic Design Considerations

8.1 OMNiMOTE Design

The major components of the OMNiMOTE circuit are the IR receivers, the IR emitters, the WIFI module, the LCD module, the power supply and the microcontroller. The major subsystems of the microcontroller that will be used are the UART, the USB interface, the SPI and the Pulse Width Modulator. Some of the main considerations in designing our circuit are choosing a clock frequency to support all the different communication protocols used and all powering modules that operate at different voltages.

8.2 Theory of Operation

Operation of the OMNiMOTE involves a fairly intuitive series of steps. These steps include configuring of the WIFI module, recording of IR commands and configuring of the user interface, and emitting IR components as a response to user interaction. An example sequence of operation is outlined below.

8.2.1 Wireless Network Settings Configuration

First, the WIFI module is configured using the USB port. When connected to a computer via an USB port, the OMNiMOTE works as a USB flash drive or mass storage device. After reset, the OMNiMOTE USB drive will contain a file listing the standard settings to which the WIFI is configured. The user can open this file and edit the entries to configure the WIFI module. After being disconnected from the computer, the OMNiMOTE enters Configuration Mode. The OMNiMOTE transmits an 'Enter Command Mode' sequence to the WIFI module via UART which sets the WIFI to command mode. The OMNiMOTE then transmits the settings stored in the configuration file to the WIFI module. The last two commands transmitted are 'save' and 'reboot' which stores the new settings in the flash memory in the WIFI module and reboots the WIFI module. [8.2] When WIFI reboots and configuration is complete, the OMNiMOTE enters idle mode. A device reset, or power off power on sequence, is needed for the network settings to take effect.

8.2.2 Virtual User Interface

If the network configuration was successful, the OMNiMOTE with connect to the Wireless network indicated by the given configurations after reset and can then receive commands via the virtual user interface. The virtual user interface is a dedicated application on another WIFI enabled device connected to the same local network that communicates with the OMNiMOTE using the OMNiMOTE protocol. As of the date of this document, he protocol is composed of 31 commands for recording IR commands, emitting IR commands and setting or getting data stored in the OMNiMOTE regarding user interface structure. More information on this protocol can be found in the software . The virtual user interface allows the user to set the OMNiMOTE in recording mode.

After detecting and recording an IR signal, the user can then configure the virtual user interface by adding buttons to the interface. The user can edit properties like the name, image, target component and location of the button, as well as assign previously recorded IR commands to a button. This information is stored on the OMNiMOTE, so any devices accessing the OMNiMOTE will be able to display all changes made via any other virtual interface

8.2.3 Recording IR Commands

When in recording mode the OMNiMOTE will wait for an IR signal from the IR receivers. The user can then point their remote at the OMNiMOTE receivers and press the button corresponding to the command he or she wishes to record. The five protocols supported are RCA, NEC, SIRC, RC-5 and JVC, the latter two being the most common among Japanese and American TVs. In order to support those five protocols, the OMNiMOTE must detect four different IR carrier frequencies, 36 kHz, 38 kHz, 40 kHz and 56 kHz. Four receivers are used to detect IR sequences in the four different carrier frequencies.

When detecting an IR signal, the OMNiMOTE stores the time elapsed between the IR signal transitions and the frequency of the signal. This will allow the OMNiMOTE to reconstruct the signal later for transmission. This data is also used to match the signal received to the appropriate protocol. The OMNiMOTE takes advantage of the differences between bit length and sequence of the five different protocols when matching a signal to a protocol.

8.2.4Emitting IR Commands

When the button is clicked in the virtual user interface, the application will inform the OMNiMOTE which will in turn emit the command to the targeted component. When executing an IR sequence, the microcontroller will use a multiplexer to select the appropriate IR emitter output. There are eight outputs in total. These are 3.5mm mono jacks that will connect to IR LEDs that point to one of the user’s components. Using the previously stored information about the IR command, the signal transitions and the protocol, the OMNiMOTE will reproduce the stored IR sequence using the PWM module.

8.2.5 OMNiMOTE Display

During the entire sequence, an LCD is used to display the status of the device (i.e. current operating mode, WIFI signal strength). The display is split in 4 31x31 pixel sections to simplify the drawing of the graphics. The first shows displays the WIFI signal strength while the other three display one of three characters representing the mode of the OMNiMOTE.

The WIFI signal strength is read from the WIFI module approximately every 15 seconds. The LCD graphic displaying the signal strength will show five solid or hollow bars. If the OMNiMOTE is not connected to a network, five hollow bars are displayed, otherwise, one to five solid bars are displayed indicating the strength of the signal, one bar for every 20% of the maximum signal strength.

The orientation of the display is detected by using a tilt contact switch tied to a GPIO pin. The display is refreshed approximately every 3 seconds, where orientation, signal strength and the mode of the device is used to adjust display appropriately. Table 8.2.1 lists and explains the different codes displayed by the LCD.

Table 8.2.1: OMNiMOTE Mode and LCD Code Descriptions

OMNiMOTE Mode	LCD Code	Description
Idle Mode	OK	OMNiMOTE is in IDLE mode, this is the only mode where WIFI signal strength information will be fetched
USB Mode	USB	OMNiMOTE is working as a USB drive.
Network Settings Configuration Mode	CFG	WiFi module is being configured. Do not power off while in this mode.
Initialization Mode	HEY	OMNiMOTE is performing standard initializations. This is the first mode the OMNiMOTE is in upon reset.
Controller Mode	CON	OMNiMOTE has established a connection with a Virtual Remote. In this mode, the OMNiMOTE will respond to button commands from a Virtual Remote Application.
Recording Mode	REC	OMNiMOTE is waiting for an IR signal from the IR receivers
Software Error Mode	ERR	Error State. Possible causes: Either the last command received was either not recognized, an invalid IR command was detected, or the WIFI module did not respond to configuration settings. Refer to user manual for more error management instructions.

8.2.6 Power Circuit

The power circuit consists mainly of a 10V DC 500 mA power supply, a 5V switching regulator and a 3.3V linear regulator. The maximum current consumed by the OMNiMOTE circuit is approximately 460mA. The components powered via the 5V regulator are the IR receivers and emitters collectively consume about 55 mA of the circuit total. The WIFI module will be powered by applying a 5V input to its unregulated voltage input. The module then produces a 3.3V regulated output that can be used to power the module. [8.3] The LTC1174-5 Switching Regulator used, can supply up to 175mA of current. [8.1] The components powered by the LTC1174-3.3 Switching Regulator at 3.3V are the WIFI module, the microcontroller, the tilt sensor, which collectively consume up to approximately 300 mA. The LTC1174-3.3 used to power them, can supply up to 425mA. [8.1] Both the LTC1174-5 and the LTC1174-3.3 can both support their respective loads.

8.3 Hardware Design Narrative

The LPC2378 microcontroller will be used as the main controller. The major subsystems of the microcontroller that will be used are the UART, the USB interface, the SPI, and the Pulse Width Modulator.

The OMNiMOTE uses an external crystal oscillator of 24Mhz to clock to provide as the input frequency for the PLL. Out of all the communication protocols used in the circuit USB requires the highest CPU clock frequency at 48MHz and is therefore the limiting lower bound when selecting said frequency.

The UART in the WiFi module can achieve standard baud rates with any crystal frequency above 2MHz. [8.3]. The WIFI module will use UART to communicate with the microcontroller. The UART baud rate is 19,200. This is baud rate is supported by both microcontroller and the WIFI module, and it’s the fastest baud rate possible without the need to implement hardware flow control. [8.2]

The pulse width modulator will be used to generate the four frequencies used to control the IR emitters; 36KHz, 38KHz, 40KHz and 55KHz. Two match registers can be used to provide a single edge controlled PWM output. One register controls the PWM cycle rate and the other register controls the PWM edge position. [8.6]. RC5 protocol requires a maximum 2% error. [8.7]. With a CPU clock speed of 48MHz, the PWM signal produces a maximum error of .0025% at 36MHz, which is within the protocol tolerance. The IR Receivers communicate with the microcontroller via GPIO ports.

The IR Emitters are driven by the PWM module in the microcontroller via an 8 bit multiplexer. Three GPIO pins are used to select the multiplexer output ports. Only one PWM output is then needed to enable the multiplexer and therefore the correct IR emitter used for output.

8.4 Summary

Using a clock speed of 48MHz the LPC2377 microcontroller will be used to control the rest of the circuit. From the different communication protocols used, USB protocol is the fastest protocol and the deciding factor in choosing a clock frequency for our oscillator. A WIFI module will provide exposure of the user interface for WIFI enabled devices and communicate with the micro controller via UART with a baud rate of 19200. The USB interface will provide access to configure the WIFI settings. The PWM in the microcontroller will generate the 4 IR frequencies supported of 36 kHz, 38 kHz, 40 kHz and 55 kHz. The LCD will communicate via general IO pins to display the status of the device to the user.

9 PCB Layout Design Considerations

9.1 Introduction

The core of the OMNiMOTE project centers around the processing, saving and outputting of IR signals and providing access to the saved command signals via wifi. In addition to processing these signals, there are also various communication interfaces that are utilized which require access to the board, including mono jacks for IR blasters and a USB type B socket. The tilt sensor, wifi module and LCD screen are internal to the board design and do not require any physical access to outside peripherals.

The primary constraint in this design is the proper layout of components such that the IR collectors will have line of sight view to any incoming signals from a remote device. Aside from this, it must also be assured that 3.3V and 5V levels will be provided for the power of components in the design. Another major constraint is the consideration of possible noise in the functionality of the wifi module, and the provision of a clear signal for the module. This will require attention when choosing casing for the finished device.

9.2 PCB Layout Design Considerations - Overall

In designing the product PCB, physical layout was first taken into account. This meant considering how the board will be physically integrated into the packaging design once completed. The OMNiMOTE will use IR collectors to read signals corresponding to button presses on remote devices and then output sampled remote commands over IR blasters using mono jacks. The reason for using multiple collectors is that different IR protocols use different signal carrier frequencies, and the OMNiMOTE will provide support for 5 separate protocols. A total of 4 frequencies will be needed to be recognized to parse signals for the 5 protocols that will be supported, requiring the use of 4 separate collectors each capable of reading a specific frequency. The collectors will be mounted to the front acrylic window of the device, and a connector will route their inputs to the main PCB. This will allow more flexibility in the mounting of the boards inside of the casing. The jacks will be mounted on the back of the device so that they are out of view. The USB jack will be located on the back of the device for the same reason.

Peripherals also need to be located as close as possible to the GPIO ports that they are accessing on the microcontroller to reduce the length of traces and prevent noise on the lines. The LCD will communicate to the microcontroller using multiple GPIO pins, including a serial data line, multiple chip selects, a data/command selection, and a clocking signal for the shift register. The tilt sensor will also require a GPIO input to the microcontroller. The wifi module uses UART to interface with the microcontroller. These communication lines won't be especially sensitive, but trace size should be as wide as possible and efficient routing should be used to ensure that excessive levels of noise do not have any significant influence on the line voltages. To help facilitate this, the microcontroller is placed in the middle of the board so that trace length to all components will be relatively similar. There are considerations that need to be taken with respect to the wifi signal. There should be no copper directly beneath the wifi module once it is soldered to the PCB. The documentation for the wifi also specifies that the integrated antenna on the module should extend at least 5mm from the edge of the PCB [9.1]. If either of these two steps are not taken, there could be attenuation of the antenna signal as a result.

Circuitry on the board can be grouped into two groups, high speed digital and analog. The microcontroller will provide a clock signal for most peripherals requiring a clocking signal, aside from the wifi module which provides its own internal clock and the LCD which requires an external 2kHz clocking signal that is provided using a 555 timer as there were no extra pin-outs to a PWM port on the microcontroller. The only handling of analog signals is done in sampling the square wave IR button command inputs from the collectors. Care needs to be taken to avoid getting excessive noise on the trace lines providing these signals to the microcontroller to ensure that they can be correctly read and encoded into memory for later output. No manufacturing constraints are foreseen, however space will be allowed for sufficient trace widths.

9.3 PCB Layout Design Considerations - Microcontroller

The microcontroller in this design will employ the use of bypass capacitors at the power input terminals. The microcontroller will be clocked relatively high at 48 MHz, and so a 0.1 μF capacitor will be used for all decoupling of the microcontroller power inputs. These capacitors will need to be located as close as possible to the power terminals on the microcontroller. As the pin pitch is small, these capacitors will be places on the other side of the PCB below the microcontroller to avoid any intrusion on other inputs to the microcontroller. As the USB functionality requires a fast, accurate clock that the microcontroller's internal oscillator is not sufficient for, a 12MHz external oscillator will be used.

Although the microcontroller provides all output logic at 3.3V levels, it supports logic input voltages up to 5V. The microcontroller needs to be properly isolated from any voltage higher than 5V as its input pins are only rated up to 5V and any voltage level higher than that will result in damage to the pin's driver. The only signals that will be sent off of the board will be the IR output signals sent to the blasters. The output jacks should be located as close as possible to the microcontroller to minimize any noise that could disrupt the signals that are being output. The power traces to the microcontroller should also be as wide as possible while also taking pin width and the power trace to pin junction into consideration. As the microcontroller will be in the middle of the board and the number of components are minimal, power trace routing to the microcontroller should not become a problem. This could become a problem depending on how small the final PCB footprint will be.

One last consideration in regards to the microcontroller is the PWM which is used to generate the output signals for the IR blasters. In order for the desired external component to respond to the IR signals generated by the microcontroller, the output signal needs to be free of noise. All of the traces from the IR output jacks to the microcontroller need to be as short and as wide as possible. As all of the jacks will be next to each other, it is possible to have noise due to electromagnetic coupling between the traces to the IR jacks. As only one component will be controlled at any one time through the use of one jack, there should be minimal to no EM coupling noise on the output line as there will be no other signals being output at the same time. This could only come into effect if signals are sent out in rapid succession.

9.4 PCB Layout Design Considerations - Power Supply

The maximum voltage needed by any component on the board is 5V which is needed by the LCD and the IR multiplexers. A 9V supply will be used to power all components on the board. This will be accomplished by using two switching regulators, one to bring the 9V power supply input to 5V and the other to bring the 5V regulator output down to a 3.3V output. All powered components on the board either require a 5V or a 3.3V level. The regulators require some manufacturer specified circuitry for them to function properly. Special care should be taken to adhere exactly to the manufacturer specified circuitry and to use the manufacturer specified components for this circuitry. This will reduce the chance of anomalies in the regulated voltages.

Trace widths also need to be as wide as possible for all power circuitry. At the very least, the 9V power supply lines should be increased as much as possible. As the board will not be excessively congested, all of the power traces should be as wide as possible. As the pins on the micro are small, the junction between the power trace and the micro pin should also be thought of. Any signal input to a pin on the microcontroller should be of sufficient distance away from a power input to allow for the greater width of the power trace.

As mentioned previously, the board will be divided into high speed digital and analog sections. The analog inputs from the collectors can essentially be treated as digital signals however. An IR collector output is tied to a GPIO pin on the microcontroller and interrupts are generated whenever that pin transitions from high to low or low to high. It is in this way that the inputs are encoded into memory. For the purposes of PCB design, all IR signals will be considered digital inputs.

In supplying power and ground to the digital ICs, as many parallel paths to ground as possible will be provided to minimize the impedance for the return signals. The copper pour for the PCB will also be connected to the ground net as well. A single point ground system will be used for all circuitry. All of the separate boards connected to the main PCB will have ground references tied to the main board. As the general layout of the board is fairly simple, there should not be any space restrictions that would become an obstacle in creating a single point ground net. This also means that there should be a fair amount of flexibility in power trace routing. This will allow power traces to be routed in parallel with ground traces.

This design will also use a variety of capacitors to reduce noise. This includes the use of bulk capacitors and bypass capacitors. At the power input terminals, an electrolytic capacitor with a value of 100µF will be used to charge the bypass capacitors used in the microcontroller and the wifi circuits. A small ceramic capacitor with a value of 0.1µF will also be used at the input for filtering of high frequency noise. These capacitors will be placed as close to the input power terminals as possible. Enough space needs to be allowed for the physical size of these components.

9.5 Summary

The primary constraint of physical layout can easily be overcome by having an effective layout plan. There are multiple available GPIO ports on the microcontroller and if done carefully, selection can be done to minimize the length of traces bridging these inputs and outputs to the peripherals they are communicating with. Due to the relative simplicity of the board design, space requirements are flexible and a single point ground system will be sufficient. Parallel lines to ground will be provided for all digital ICs. Use of bypass and/or bulk capacitors where needed in the design will limit the amount of coupling noise, further reducing the chance of EMI related problems occurring.

10 Software Design Considerations

10.1 Introduction

While the OMNiMOTE does have significant hardware design, the majority of features will be implemented in software. For example, we will be decoding infrared commands in software rather than trying to find an integrated circuit capable of decoding infrared protocols. This makes the software more complicated, but allows the hardware to be much simpler. The major software components that need to be created include the USB interfacing, infrared encoding and decoding, LCD interfacing, and Wi-Fi communication.

10.2 Software Design Considerations

The overall software design will be interrupt-driven. Since the device will be on at all times, we need to be able to save power as much as possible. If we use interrupts, we can allow the device to sleep during periods of extended inactivity. Then, when interrupted by a Wi-Fi command, it can wake up, handle the task, and then go back to sleep. The only interface to the device will be through the client application, so the only interrupt we need to watch for when sleeping is the Wi-Fi data interrupt.

Since we are developing on a microcontroller, we will not be using any kind of memory allocation. Instead, we will create static variables for any memory we require and modify them when needed. This is a little inefficient, since we will have memory allocated even when we aren't using it, but much easier to work with than trying to dynamically allocate memory while running. Table 10.2.1 shows the major memory components and their corresponding sizes.

Table 10.2.1 – Memory Usage

Name	**Data Size (Bytes)**	**Number of Items**	**Total Size (Bytes)**
IR Command Table	6	127	762
IR Input Arrays	2	280	460
LCD Graphics	120	35	4200
Components	20	8	160
Buttons	29	127	3683
IR Signal Information	6	14	84
IR Signal to Send	283	1	283
Total Memory			**9632**

The JTAG pins on the micro have been connected to a header on our PCB for programming and debugging. Not only will this allow for firmware updates later on in the development cycle, but it will allow us to step through our code very easily while the micro is soldered onto the PCB. There will also be limited debugging routines that the main program will use to make sure everything is working properly without needing the JTAG connected. If it finds an error, an error code will be displayed on the LCD display.

10.3 Software Design Narrative

10.3.1 Core Library

The core library (OmnimoteFoundation) contains all the enumerations, types, and definitions that will be used in all other code modules. This is fairly simple, and will be inherited by every module used in this project.

10.3.2 Interface Library

The interface library (OFButton, OFCommand, OFComponent, OFController) contains all the information that is used to create and store the user interface. It will store all the created buttons, commands, components, and controllers. A command corresponds to a single infrared signal. A button corresponds to an entity on the interface that will send one or more commands. A component corresponds to a device you are controlling and keeps track of which port associates to which device. A controller corresponds to a list of buttons, allowing you to organize the entire interface based on function, device, or any other means.

10.3.3 Infrared Library

The IR library (OFIRComponent) holds all the information required to setup and send IR signals. The function to send an IR command is sendCommand(). This function takes command index as its parameter. It locates the index in the command array, and uses the information to build an IR signal. The IR signal is stored as an array of "pulses," where each "pulse" is defined as a length of time in which the IR output is either modulating or not modulating. A cycle counter will be used to keep track of how long a pulse has been on. Once this signal has been created, the sendCommand() function starts the PWM output and enables the interrupts. At this point, the output will run entirely on interrupts, allowing the microcontroller to go back to the polling loop and service other modules.

When the PWM interrupt is called, it decreases the pulse counter by one. If the pulse counter is greater than zero, then the output doesn't need to change yet and the method just returns. If the cycle counter is 0, it will get the next "pulse," reset the pulse counter to the appropriate value, set the duty cycle to the required amount, and then return. If there are no more pulses left, it simply turns off the PWM output and sets a few flags to let the main loop know that it is finished.

10.3.4 Wi-Fi Library

The Wi-Fi library (OFMessage, sio) contains methods and information regarding communication with the Wi-Fi module. The Wi-Fi module communicates through UART operating at a baud rate of 115200. When the micro receives data, it will trigger an interrupt which loads the data into memory. This will then set a flag to let the main loop know that data is available to be read.

The main loop will parse the message, take the appropriate action, and send a response to let the user know that it has received the message.

Timer0 will check our UART transmit buffer every 50 ms to see whether there is any data to send. If there is, it will write the bytes in the buffer to the micro's UART transmit register to actually send the data. There are flags differentiating a full message from a partial message and a command message from a data message. We differentiate full and partial messages so that we can write to the transmit buffer multiple times and still have it send a single message. We differentiate command from data messages because the Wi-Fi module requires different formats for each.

10.3.5 LCD Library

The LCD library (OFLCD) contains everything required to run the LCD display. It handles all refreshing, displaying icons, and displaying text associated with the display. There will be an independent timer running at a 3 second interval that will be used to update the LCD display. This isn't the only time the display will be updated, but it will allow the display to check its orientation and determine whether or not it needs to re-orient the display pattern. This library will also contain interfacing functions to allow other modules to change the text or icons on the display. Once another module changes the display, it will not be displayed until the next timer interrupt.

10.3.6 USB Library

The USB library (usbcore, usbdesc, usbhw, usbuser, DiskImg, mscuser) will be used to communicate with a PC via the USB port. This library will essentially be a wrapper for the Keil USB library [10.1] that is on the Keil website. When the device is plugged in to a PC, this library will model the OMNiMOTE as a mass storage device. A program on the PC can then place a configuration file into the OMNiMOTE's memory to use on the next reboot. This feature is only used to change settings when the device cannot connect to a network. Otherwise, most settings will be able to be changed through the web interface.

10.3.7 Flash Memory Library

The flash memory library (FlashPrg, FlashDev) will be used to save settings to the OMNiMOTE in non-volatile memory. This allows the IR commands, buttons, components, and other settings to be recalled even after power is removed. This isn't necessarily a huge deal, since the device will most likely be on at all times, but it will help for those few situations that cause power to be removed. It would not be very convenient if you had 80 buttons programmed and lose it all because a storm knocked out your power.

10.4 Summary

The code has been designed to be modular. Since there were four people all working on the same project, we wanted to make it able to work on our respective modules in parallel. Once we each finished our parts, we could work on integrating it into the main project code. While this ended up making development a lot faster, we did run into some trouble trying to bring everything together. Functions would be slightly different, items would be declared multiple times, registers would be overwritten. Overall, though, writing the code in parallel made development a lot faster.

Adding JTAG support and debug routines helped a lot in developing as well. Our code could run perfectly on the development board but still crash when we run it on our PCB. Having something to step through the code allowed us to easily figure out what was wrong. Keil's IDE also helped a lot in debugging programs. You can view pretty much any register or memory location, and it even has extra windows for each individual subsystem in the microcontroller.

11 Version 2 Changes

OMNiMOTE 2.0 would have many improvements to make it easier to program, easier to build, and easier to use. First, we would have many pre-installed commands from the 2 most common protocols (RC5 and NEC), that way users would not have to make it learn every command separately. We would also implement the web server we initially wanted to implement, truly making it compatible with most web devices, instead of using our dedicated apps. The construction of the OMNiMOTE would also be changed so that the front and back are molded from a single piece of acrylic or molded plastic, and the aluminum would be custom bent to match the shape perfectly.

12 Summary and Conclusions

In this project, we learned how to separate the work of a major project efficiently, collaborate on work as we start to get closer to merging our efforts, and work together to help solve issues that plagued our design. This is the first time most of us in the group have really designed a product from nothing, soldered surface mount components, or connected this many components together in an embedded design. As we were progressing through the semester, we realized some of our ambitions were too complicated to do in the given time, or too complicated for us to do at all without knowing a lot more information. This led to features we wanted being put onto the chopping block, and as time became critical, those features were cut. The web server was going to be difficult with our current hardware, as well as some protocols that would have been difficult to control, like actions when holding down a button versus pressing. These were cut since we could still have a lot of functionality, even without them. Some parts were too hard to integrate, like the OLED display. It used a flexible cable connector which was supposed to be soldered onto a PCB, but ended up getting damaged beyond repair while the soldering was attempted.

Knowing what we know now, we would most likely have tackled the project a little differently, but there would still have been unforeseen issues to plague us. We learned a lot of information from this experience, teaching us how to apply the knowledge we learned throughout our ECE careers onto a real-world device. We were able to learn how to troubleshoot issues, whether they were hardware, software, or both, and that helped recognize solutions faster as we went along.

13 References

[3.1] Lantronix WiPort Data Sheet. March 2006 [Online]. Available: http://www.lantronix.com/pdf/WiPort_DS.pdf [Accessed September. 2010]

[3.2] PIC32MX Family Data Sheet. 2008 [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/PIC32MX_Datasheet_v2_61143B.pdf [Accessed September. 2010]

[3.3] Roving Networks WiFly GSX Data Sheet. August 5, 2010 [Online]. Available: http://www.rovingnetworks.com/documents/rn-131-ds.pdf [Accessed September. 2010]

[3.4] STM32F103x8 Data Sheet. 2010 [Online]. Available: http://www.st.com/stonline/products/literature/ds/13587/stm32f103c6.pdf [Accessed September. 2010]

[3.5] ThinkFlood RedEye Product Page. 2010 [Online]. Available: http://thinkflood.com/products/redeye/ [Accessed September. 2010]

[4.1] Doyle. "Infrared Repeater" U.S. Patent 5 602 664, Feb. 11, 1997.

[4.2] Schlarb. "Connecting Infrared (IR) Controllable Devices to Digital Networks" U.S. Patent Application Publication 0020999, Jan. 6, 2006.

[4.3] Belz at al. "Virtual Universal Remote Control" U.S. Patent Application Publication 0134338 A1, Jun. 3, 2010.

[5.1] United States Department of Defense, "Military Handbook Reliability Prediction of Electronic Equipment," December 1991. [online]. Available:

https://engineering.purdue.edu/ece477/Homework/CommonRefs/Mil-Hdbk-217F.pdf. [Accessed: November 11, 2010].

[5.2] NXP Semiconductor, "LPC2377/78 Product Data Sheet", June 2010. [online]. Available: https://engineering.purdue.edu/477grp3/datasheets/MICRO%20-%20LPC2377_78[1].pdf. [Accessed: November 11, 2010].

[5.3] Linear Technology, "High-Efficiency Step-Down and Inverting DC/DC Converter". [online]. Available: https://engineering.purdue.edu/477grp3/datasheets/power/1174fe[1].pdf [Accessed: November 11, 2010].

[5.4] NXP Semiconductor, "Reliability Qualification Report: LPC23xx and LPC24xx Rev. D Microcontroller Products", May 2009 [online]. Available: https://engineering.purdue.edu/477grp3/datasheets/LPC23XX%20Reliability.pdf. [Accessed: December 11, 2010].

[5.5] Newhaven Display, "NHD-12232KZ-NSW-BBW-P". May 2010 [online]. Available: https://engineering.purdue.edu/477grp3/datasheets/LCD%20-%20NHD-12232KZ-NSW-BBW-P.pdf [Accessed: December 11, 2010].

[5.6] Avant, "SBN1661G_M18, SBN1661G_M02, SBN0080G_S18, SBN0080G_S02", July 2007 [online]. Available: https://engineering.purdue.edu/477grp3/datasheets/Avanti SBN1661G[1].pdf. [Accessed: December 11, 2010].

[6.1] "RoHS Legislation and Technical Manual," Newark InOne, 2005. [Online]. Available: http://www.newark.com/pdfs/RohsTechManual_v2.pdf. [Accessed: Nov. 18, 2010].

[6.2] Roving Networks, "“WiFly GX” 802.11G Super Module," Roving Networks. [Online]. Available: http://www.rovingnetworks.com/documents/rn-121g-ds.pdf [Accessed: Nov. 18, 2010].

[6.3] Elizabeth Wasserman, "How to Dispose of Computers, Cables, Keyboards, Mice and More," earthlinksecurity.com, 2007. [Online]. Available: http://www.earthlinksecurity.com/articles/Ewaste/index.html [Accessed: Nov 18, 2010].

[7.1] BitWise Controls Website. 2010 [Online]. Available : http://www.bitwisecontrols.com/ [Accessed September 2010]

[7.2] ThinkFlood Products Website (RedEye). 2010 [Online]. Available: http://thinkflood.com/products/redeye/ [Accessed September 2010]

[7.3] Engineers Edge (Weight of Acrylic). 2010 [Online]. Available: http://www.engineersedge.com/wwwboard/posts/15096.html [Accessed September 2010]

[7.4] Answers.com (Weight of Aluminum). 2010 [Online]. Available: http://wiki.answers.com/Q/What_is_the_density_of_aluminum [Accessed September 2010]

[8.1] Linear Techonology. (2010, 6, 17). High Efficiency Step-Down and Inverting DC/DC Converter [Datasheet]. Available: http://www.google.com/url?sa=t&source=web&cd=2&ved=0CBYQFjAB&url=http%3A%2F%2Fcds.linear.com%2Fdocs%2FDatasheet%2F1174fe.pdf&ei=Si21TKeFFIGBlAed_KScCw&usg=AFQjCNFDWAe9OYKsODuRaFtADlw9k7wcsA&sig2=NEM4Hf9BiIKynljsey_cHw

[8.2] Roving Networks. (2010, 5, 24). "WiFly GX" Case Study [Datasheet]. Available: http://www.rovingnetworks.com/documents/Wifly-GSX_Case%20studyData%20logger.pdf

[8.3] Roving Networks. (2010, 5, 24). "WiFly GX" 802.11G Super Module [Datasheet]. Available: https://engineering.purdue.edu/477grp3/datasheets/WIFI%20-%20rn-121g-ds%5b1%5d.pdf

[8.4] STMicroelectronics LIS302DL [Datasheet]. Available: http://www.st.com/stonline/books/pdf/docs/12726.pdf

[8.5] NXP. (2010, 6, 17). LPC2377/78 [Datasheet]. Available: https://engineering.purdue.edu/477grp3/datasheets/MICRO%20-%20LPC2377_78%5b1%5d.pdf

[8.6] sbprojects. (2009, 09) Phillips RC5 Protocol [Datasheet]. Available: http://www.sbprojects.com/knowledge/ir/rc5.htm

[8.7] Vishaysemiconductors. IR Receiver Modules for Remote Control Systems [Datasheet]. Available: https://engineering.purdue.edu/477grp3/datasheets/IR%20-%20tsop22xx.pdf

[9.1] Roving Networks, "WiFly GX 802.11G Super Module" Roving Networks. [Online]. Available: http://www.rovingnetworks.com/documents/rn-121g-ds.pdf. [Accessed: Sep. 21, 2010].

[9.2] NXP, "LPC2377/78" NXP. [Online]. Available: http://www.nxp.com/documents/data_sheet/LPC2377_78.pdf. [Accessed: Sep. 21, 2010].

[9.3] Vishaysemiconductors, "IR Receiver Modules for Remote Control Systems" Vishaysemiconductors. [Online]. Available:

https://engineering.purdue.edu/477grp3/datasheets/IR%20-%20tsop22xx.pdf. [Accessed: Sep. 21, 2010].

[9.4] Linear Technology, “LTC1174/LTC1174-3.3/LTC1174-5 – High Efficiency Step-Down DC/DC Converter” Linear Technology. [Online]. Available: http://cds.linear.com/docs/Datasheet/1174fe.pdf. [Accessed: Oct. 2, 2010].

[10.1] Keil, "LPC2368/LPC2378 USB Mass Storage Device Example" Keil. [Online]. Available: http://www.keil.com/download/docs/336.asp. [Accessed: Dec 7, 2010].

Appendix A: Individual Contributions

A.1 Contributions of Matthew Barga:

These are the contributions of Matthew Barga in the development, implementation, construction and debugging of the Omnimote design. In development, Matt's contributions included input into initial decisions on major component selection and design of circuitry for power supply and USB functionality. He was also in charge of all wifi functionality and integration in the initial stages. In implementation, Matt helped to debug hardware once it had been soldered to the PCB, and helped prototype circuits before the associated components were soldered to the final PCB.

In the Omnimote PCB and schematic design, Matt created the footprints and decals for most of the components. He gained a lot of experience in using the PADS software and helped other teammates when they had any questions relating to the software. His main contributions to the schematic design were the allocation of microcontroller ports for all components accessing the microcontroller, design of the USB interfacing circuitry and the rendering of all IR components and the wifi module into the schematic software. In the later stages of the PCB design, he helped in component placement, trace routing and in the debugging of possible manufacturing errors. In addition, his idea to pin out extra inputs to the microcontroller for use in case the backup LCD needed to be used in place of the primary LCD proved useful in the final design.

In component selection, Matt chose the microcontroller to use for the application. He also decided on the wifi module that ended up in the final design. He played the initial role in determining an appropriate power circuit as well, and found the switching mode regulator model that ended up being used in the design. When the LCD circuit board was developed, Matt prototyped the +5V to -5 voltage inverter using 555 timers and the negative voltage regulator circuits that Nate had found and integrated them to produce the -1.5V reference voltage for the LCD contrast. He also integrated the 8-bit shift register into this circuitry.

In software, his initial contributions were the UART interrupts and buffer routines. These were used for all communication with the wifi. He did a lot of research into the software and hardware interfacing of the wifi, and developed the first communication routines on the micro for this purpose. He also imported USB functionality into the main software, and made the decision to use mass storage as the USB function to import settings from a PC for upload of network settings to the wifi over UART. Matt also played a large role in developing LCD display routines. He did the initial research into the hardware interface of the LCD and determined how to send display data to the LCD. He helped find example application code and was successfully able to modify it with the help of Alex Reyes to fit the needs of the OMNiMOTE design.

In addition to all of the technical contributions listed above, Matt also provided some assistance in other duties. He helped to a limited degree in finding and purchasing components at manufacturers' websites. Matt also did a lot of literature lookup for component datasheets. He also provided the initial steps in setting up the team website, and did occasional maintenance to improve the layout of the site.

A.2 Contributions of Nathan Meyers:

As team leader, I helped organize our ideas at the beginning; as well as assign each person to a section that they are strongest with. Being the weakest at embedded code design, I was in charge of making the packaging, the controller functions, and a lot of the supporting roles.

Early on in the semester, I brainstormed different ideas for the packaging coming up with 4 different designs. I designed them in Google SketchUp, and showed them to my group members to get their opinion, and modified the best design to the current design. I built a more detailed version of the design in SketchUp, and started working on the specifics. I wanted to use a CNC Mill to mill the back panel's holes for all the components, so I used AutoCAD 2011 to design the panels with the exact specifications as all of the components that would be in them. To get the parts milled, I went to the Purdue Artisan and Fabrication Laboratory. They required the designs be from software called CaTIA, so I redrew the designs using my AutoCAD files as reference. In this software, I updated the drawing to include some of the useful structural supports like the slit that holds the USB PCB in place. I spent time checking on the progress and getting most of the parts for the packaging done. Once all of the parts were done, I worked on assembly of the packaging until it was completed.

Along with the packaging design, I was also in charge of the human interaction aspects of the project. At the beginning, this involved researching how web servers functioned, as well as learning Javascript. As we started working on the with the WiFi module, we believed that the WiFi module wouldn't be able to work well with the HTTP protocol, so this was scrapped in place of the dedicated app. I, at this time, began looking into Android development when we decided to just go with C#, since the embedded code was coming along and needed to start being tested. With C# being my strongest language currently, I first worked on the backbone of the app, like button functionality, and GUI. Once Alex had given me his first draft of his protocol I would use, I wrote separate functions for each command in the code so it was easier to read and edit if needed. As I started using the protocol, I was able to collaborate with Alex to make the protocol stronger. I then was able to use my software to help debug the embedded code as we were finishing the project.

As for the rest of what I did, I contributed a lot of the User Manual, the poster, and I am in charge of filming and editing the video. I also helped with any of the work that needed to be finished up, like de-soldering the pitch convertors from the WiFi module, finding parts, finding example circuits online to use, and soldering some of the parts on the board. I also came up with some of the improvised solutions to some of the issues while constructing the final design, like switching the Factory-Reset switch on the PCB (which was implemented as active low instead of active high) to the jumper, as well as the covering aluminum material, and placing rubber feet on the PCB tray to prevent shorting the board.

A.3 Contributions of Alex Reyes:

This section lists the contributions made by Alex Reyes's contributions on the development, implementation and debugging stages of the project. In the development stage, Alex Reyes's contributions included taking part in the overall schematic design, the creation PCB decals and the design of the software structure. In the implementation stage, Alex was in charge of soldering of all of the components and involved in the debugging of most of the circuits.

When collaborating on the overall design of the OMNiMOTE schematic, he was primarily in charge of the power supply circuit. This included creating the schematic and PCB decals for all components necessary in all three voltage levels, 9V, 5V and 3.3V.

Another significant contribution in this stage was the idea to split up the schematic design into one main board and three other peripheral boards, an IR Receiver board, an LCD board and a last board composed of the USB circuit and the IR emitters. This greatly helped to smooth out the integration stage of our design. We were able to test each board individually and make changes in a single board without affecting the rest of the circuit.

On the software side, Alex developed foundation files that were used for the main structure of our program. These are .h and .c files that contain methods, structures and definitions for Buttons, Controllers, Components, Images and Commands. He also constructed table to map out image references to an actual display mode and image to a button. In addition, he designed a communication protocol for applications to communicate with OMNiMOTE device via WIFI. He wrote and debugged most of the functions necessary to receive messages and send the corresponding replies.

He created the graphics for the LCD display for the five states of the WiFi strength symbol, the letters required to spell out the state of the device and the 90 degree clockwise rotation of all the images. He also developed functions to parse the graphics and refresh the screen. Alex developed Mac and Windows Applications that were used to obtain the WLAN settings from the user.

Alex also worked on code to configure the WiFi module via UART. The functions send a command via UART to set the WiFi module to and remove it from command mode, then wait for the WiFi module's response.

From a project management perspective, Alex contributed by creating and maintaining regular spreadsheets, list and databases to help the overall organization of the team. These documents included a project task list that detailed the nature, description, status and due date of steps in the projects. Another useful tool was a spreadsheet database of all of the components used in the project archived in the group account. He also kept account of all of the purchases made and expenses incurred by each one of the members to ensure proper reimbursement at the end of the semester.

A.4 Contributions of Joe Riley:

From the start of the project, I was put in charge of the infrared aspects of the project. I didn't know a whole lot about infrared protocols, so my first task was to research how they worked and brainstorm ideas on how we could implement them. I found a very helpful website called SBProjects that had an infrared section dedicated to different IR protocols. Using this web-site, I was able to determine which protocols we were going to support.

Once I determined the protocols, I could start researching the parts I needed in order to implement these protocols. For starters, I knew that we would need four different infrared receivers since were supporting protocols that used a total of four different carrier frequencies. This also required us to generate four different carrier frequencies, which I initially tried using 555 timers to do. After some feedback form course staff, I decided PWM would be easier and more economical, so I updated the output schematic to use PWM instead of a 555 timer.

After some initial testing making sure the input and output circuits would work, I started helping to develop the PCB. I didn't help very much in the initial stages, but as time went on I started to do more and more for the PCB design. After we got our PCB back from the manufacturer, I was looking over the board and noticed that the microcontroller footprint looked weird. After further investigation, I noticed that the footprint we created was wrong. I told our team and we started working on redesigning it to work. Alex and Matt did most of the redesign, but when I came in the next day there was something wrong with the file they were using. I asked the TA and he recommended we just start from scratch again, which I did. After a few hours I had Matt and Alex's design placed again and working.

While we were waiting for our second PCB, I started working pretty hard on the software that would control the infrared interfaces. I started out writing code that could output an infrared signal given a binary command and a protocol. After debugging it on our development board and verifying that the signal seemed correct, I started looking into how I would decode an infrared command. I noticed that all the infrared protocols we supported had a start bit that lasted for different lengths of time. I decided that the start bit would be the distinguishing factor for determining what protocol is being sent. After some thought, I also realized that we would need the processing to be interrupt driven, so we don't end up taking up all the processor time.

After converting the input and output to be interrupt driven, and verifying that the input and output signals seemed to be correct, I took the development board home with my to verify the signal could actually control a TV. The first run didn't work, but after fixing the problems back in lab I brought it home a second time and successfully controlled my TV with our development board. The infrared encoding and decoding was now complete, for the most part. Designing and developing the infrared circuits and software was by far my biggest contribution to this project, since I was really the only one working on it.

Once most of the infrared software was done, I integrated it with Alex's overall structure and started working on whatever needed to be done at the time. This included software debugging for other code modules, packaging design and creation, and PCB debugging. I was not the primary source for any of these projects, but more of a secondary helper to whomever was currently working.

Appendix B: Packaging

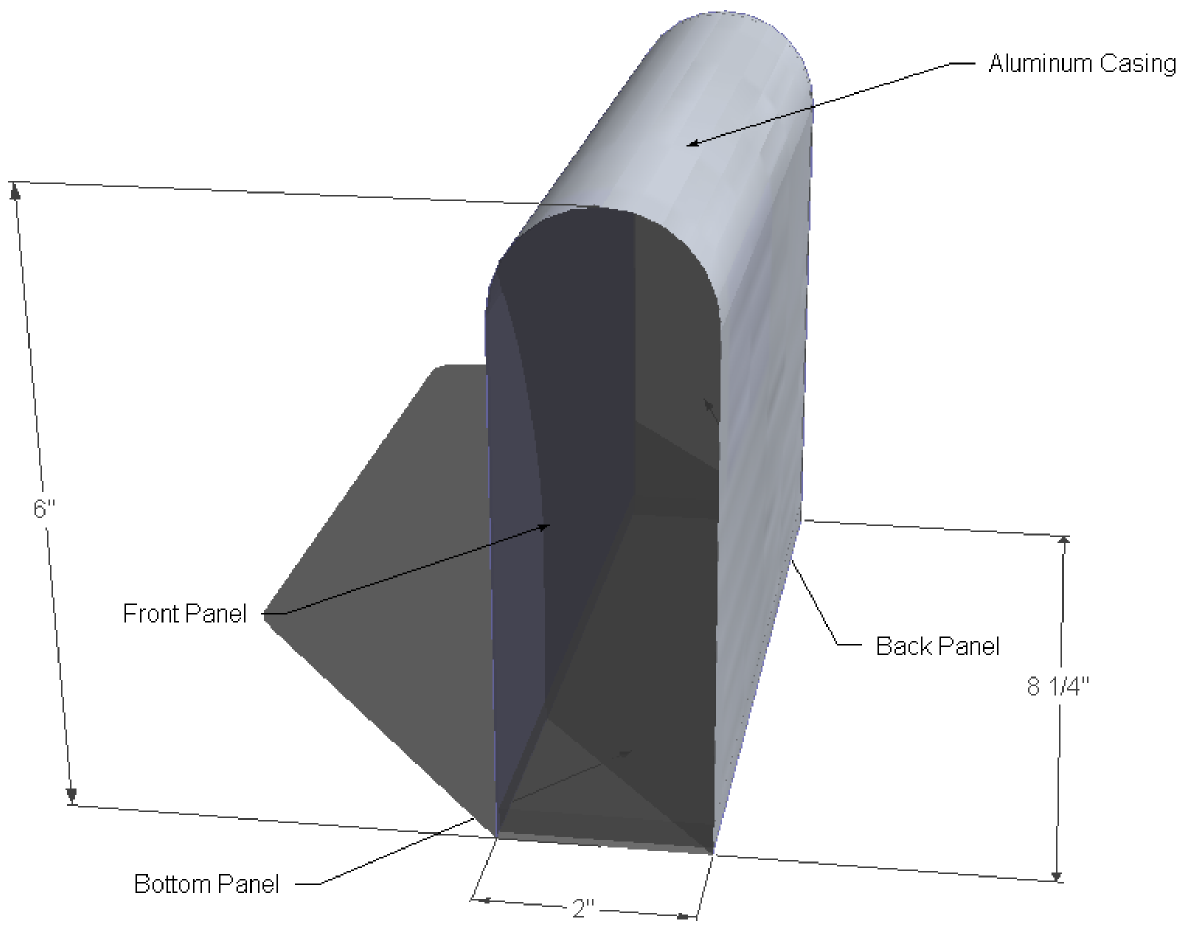

Figure B-1: Isometric View of Casing

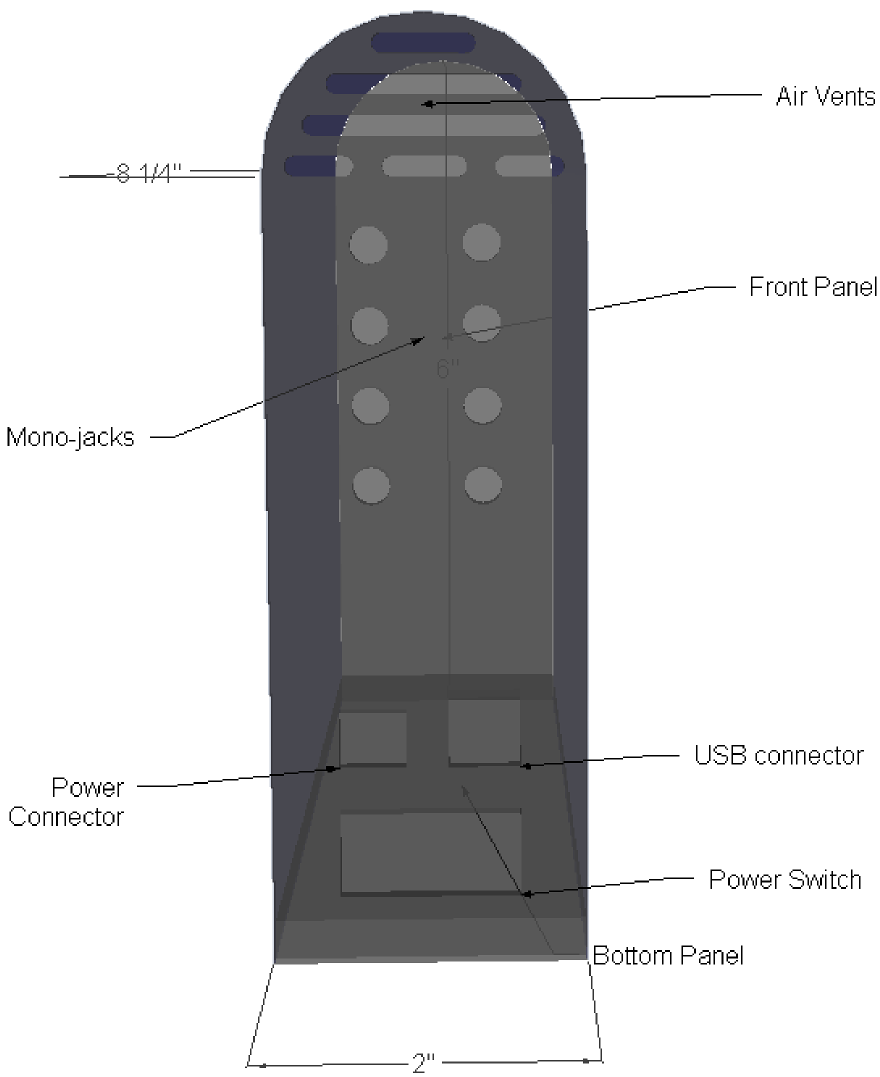

Figure B-2: Back Panel of Casing

Appendix C: Schematic

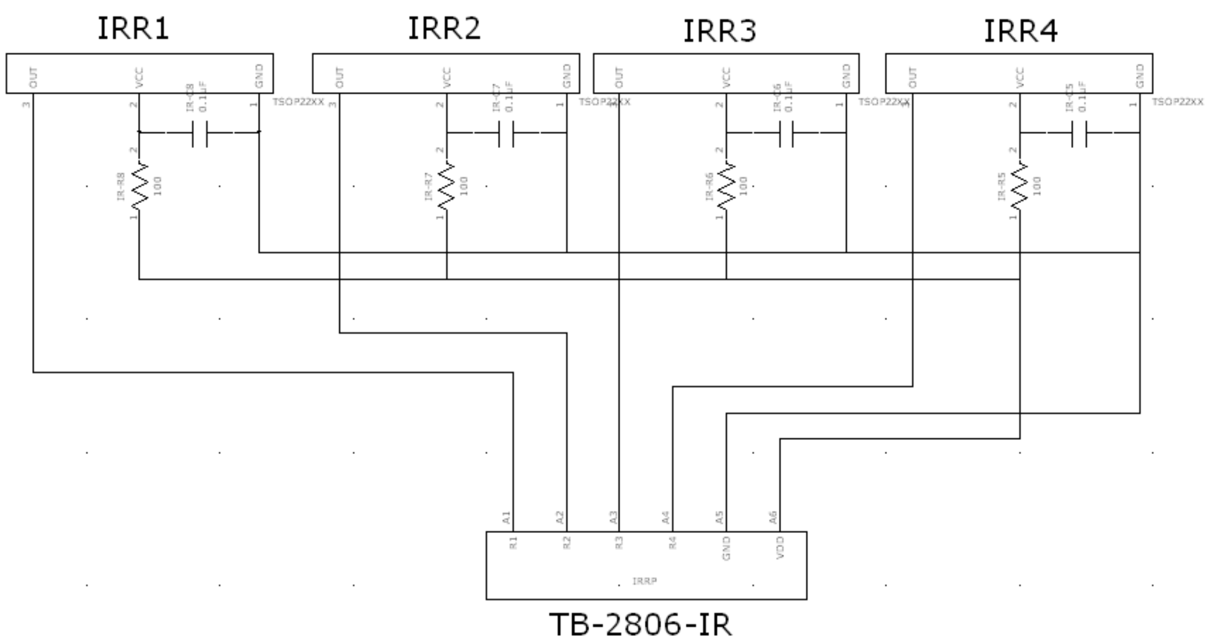

Figure C-1: IR Receivers

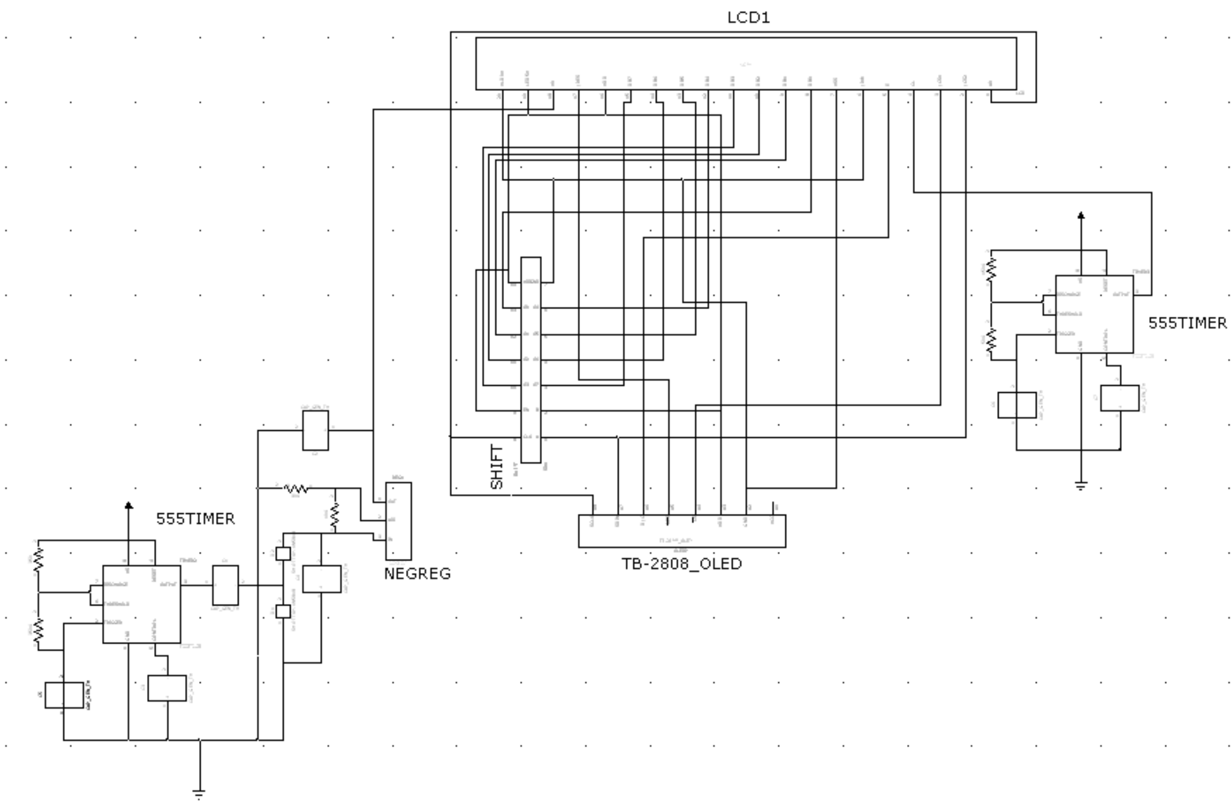

Figure C-2: LCD Display

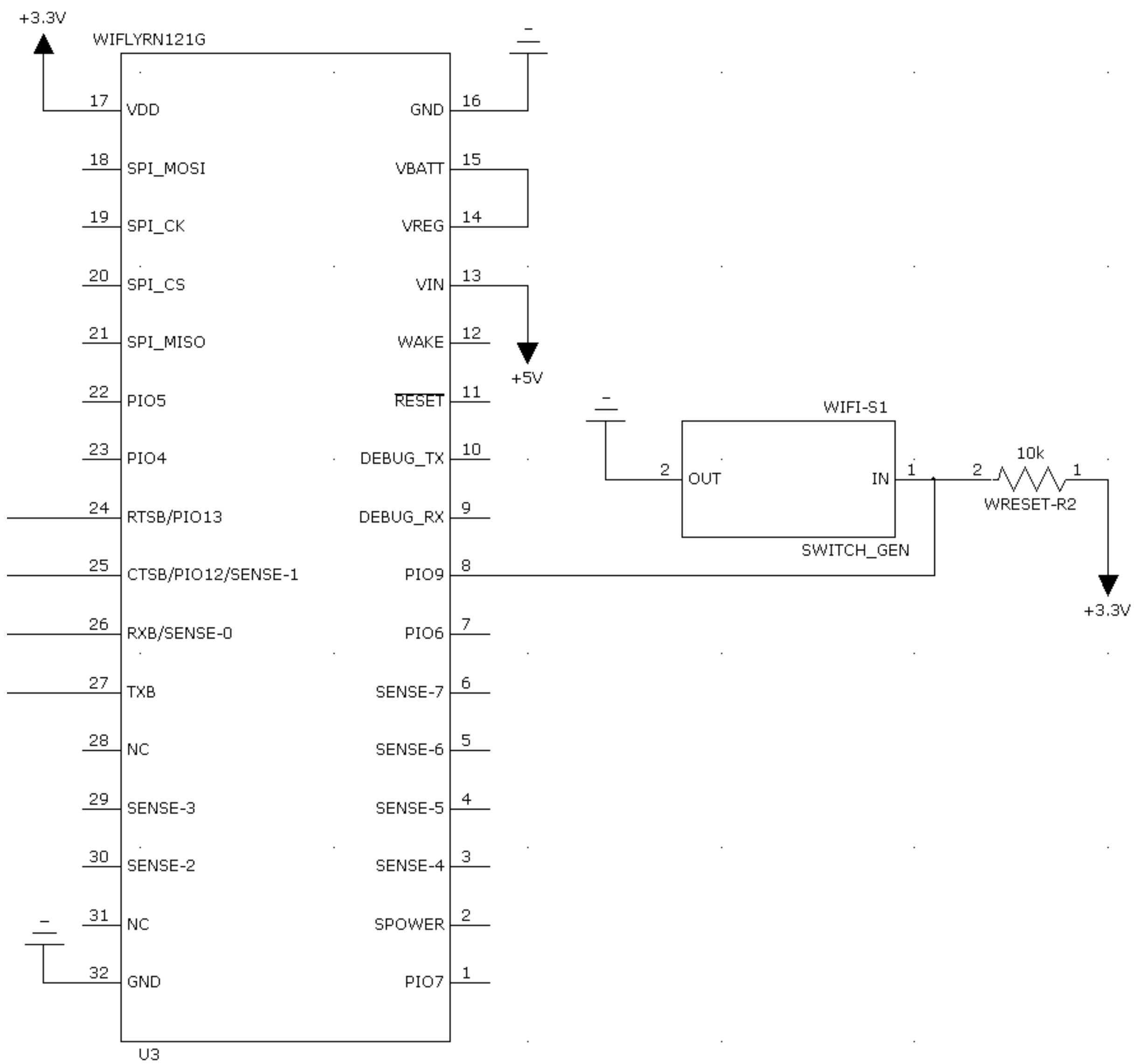

Figure C-3: Wi-Fi Module

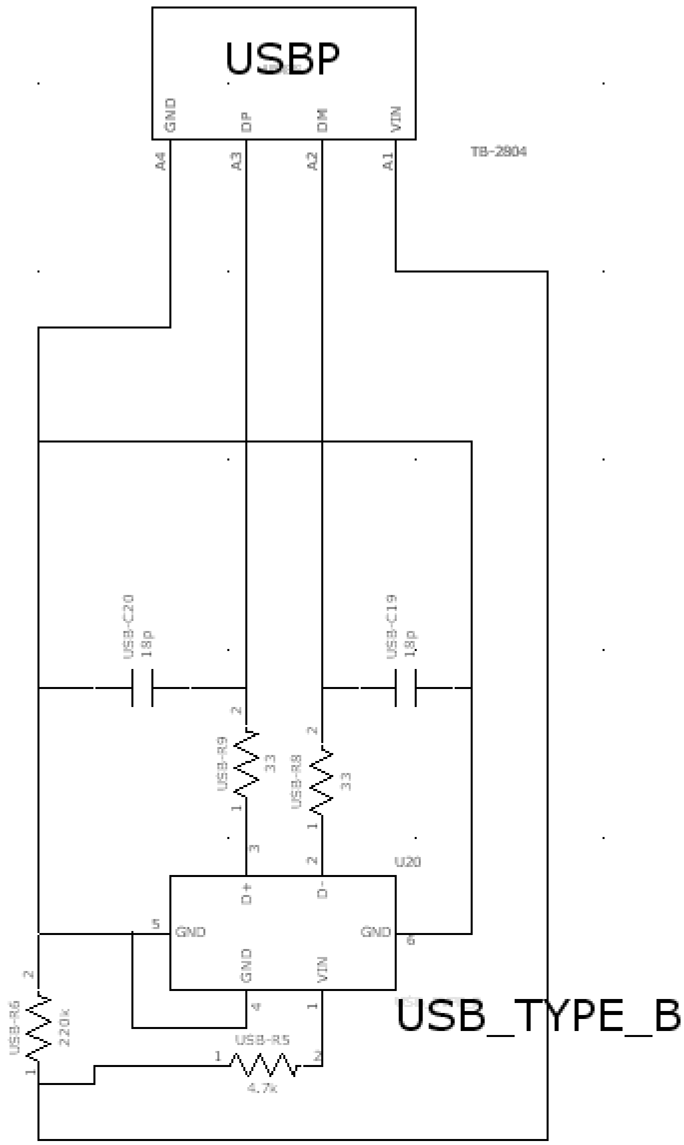

Figure C-4: USB Module

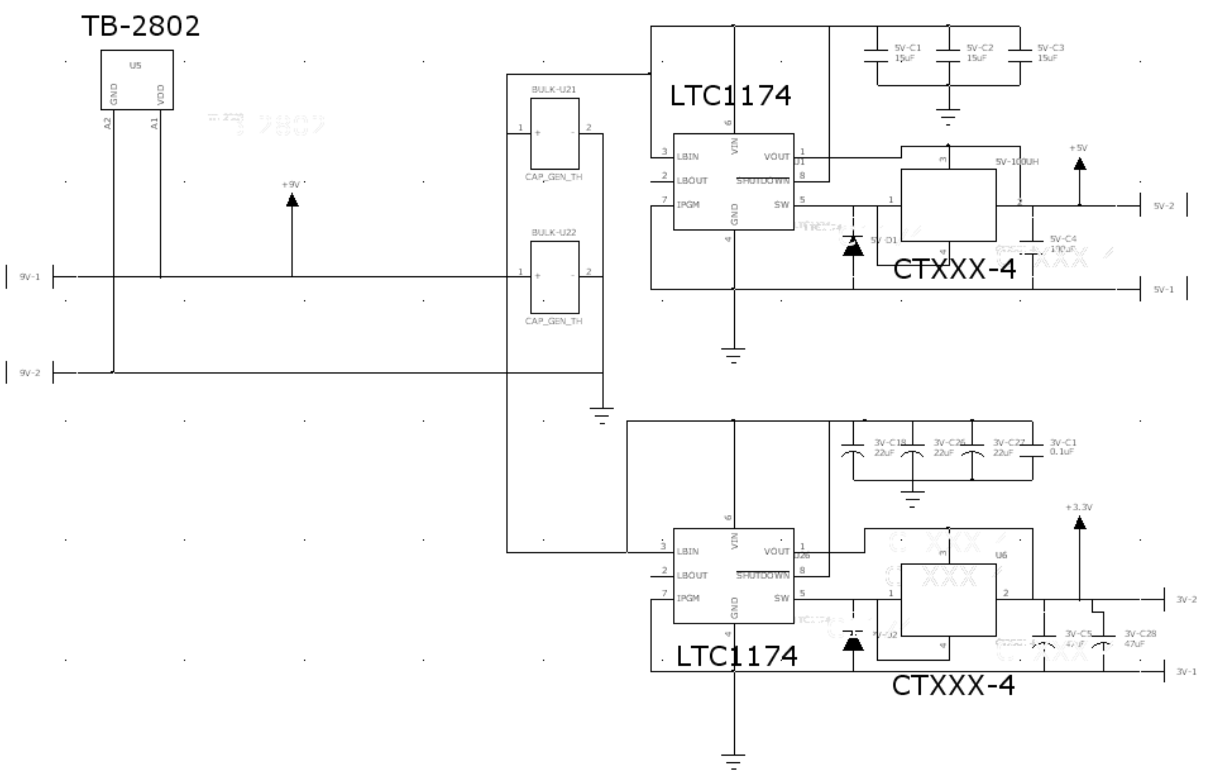

Figure C-5: Power Supply

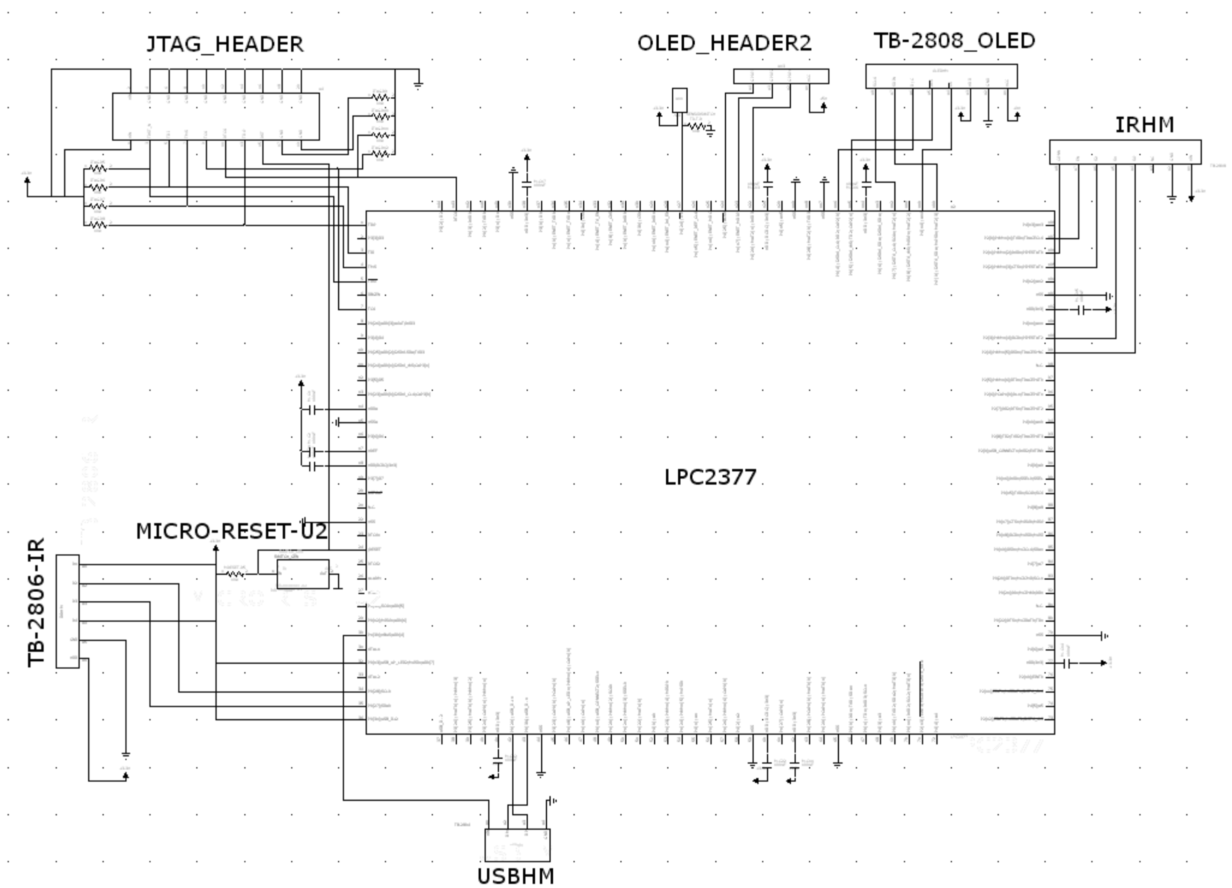

Figure C-6: Microcontroller

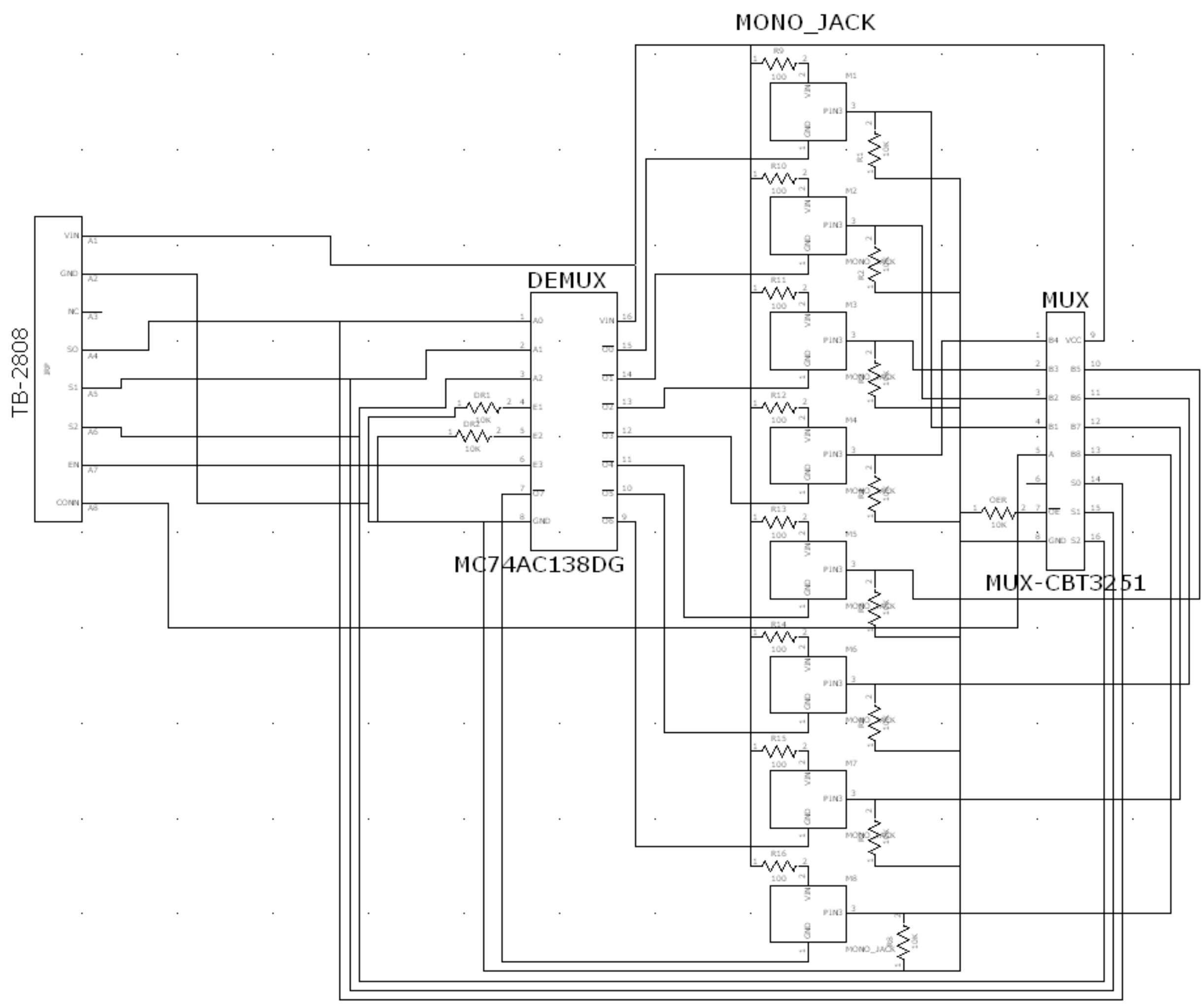

Figure C-7: IR Emitters

Appendix D: PCB Layout Top and Bottom Copper

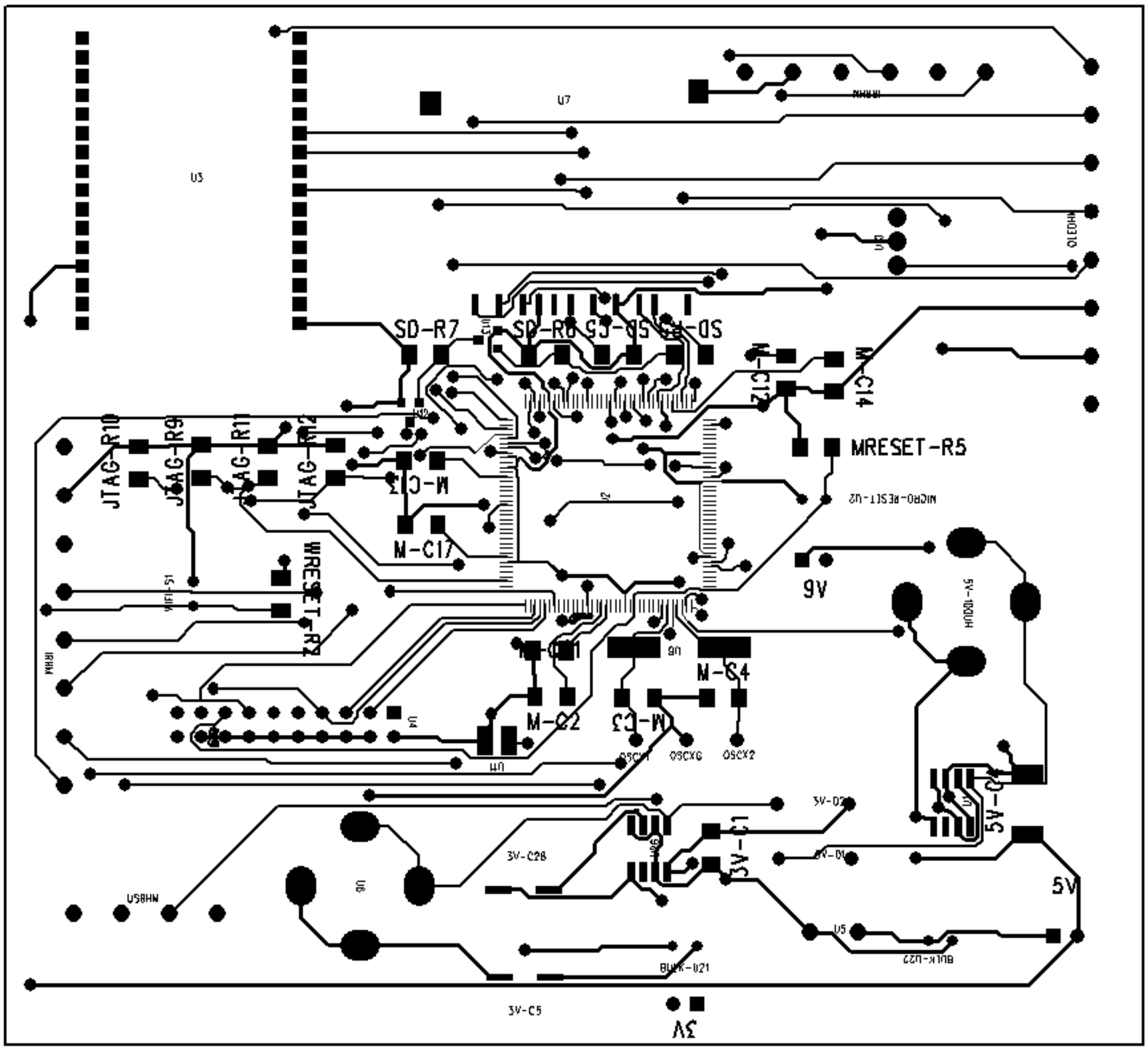

Figure D-1: Main Board Top Layer

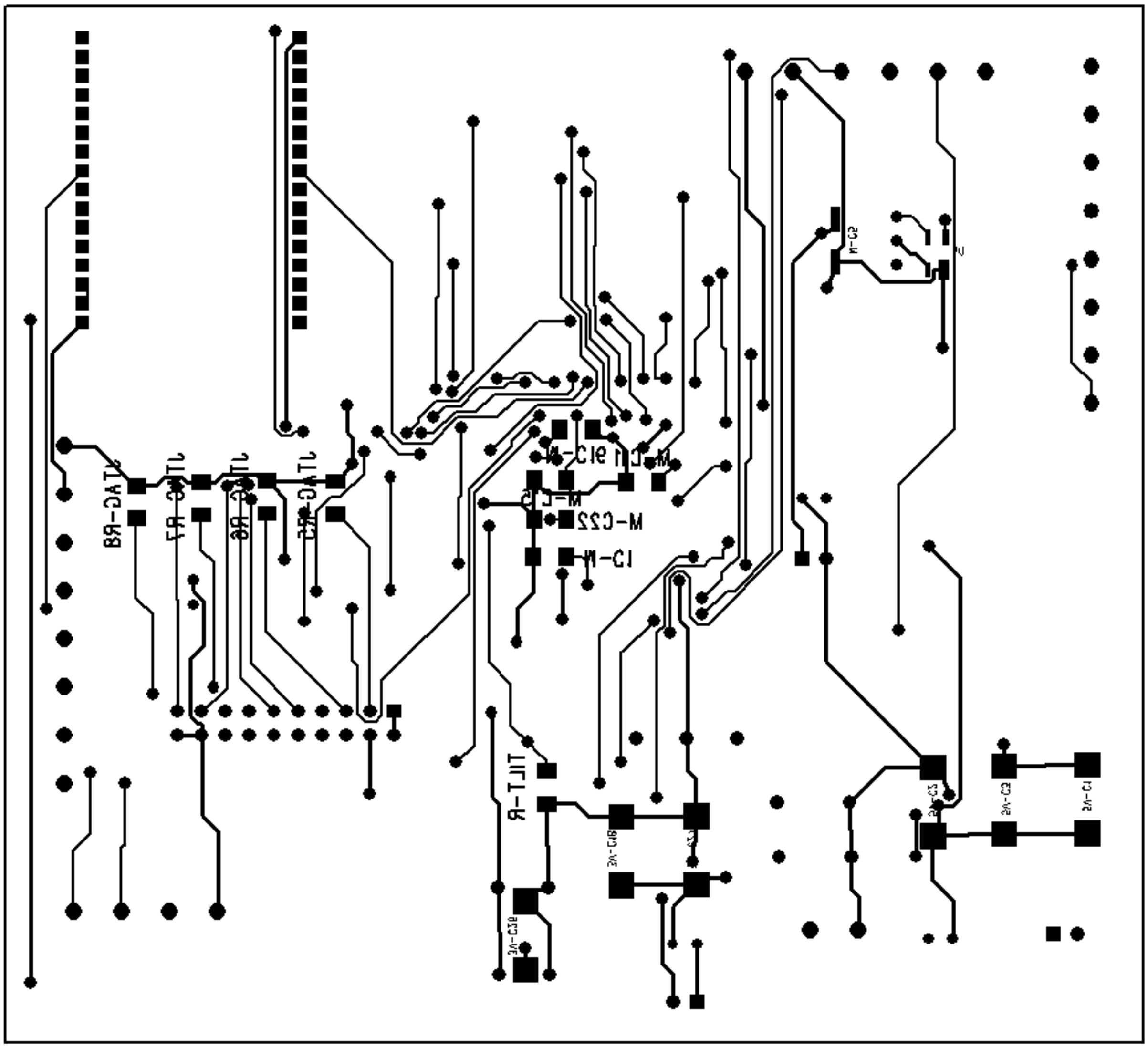

Figure D-2: Main Board Bottom Layer

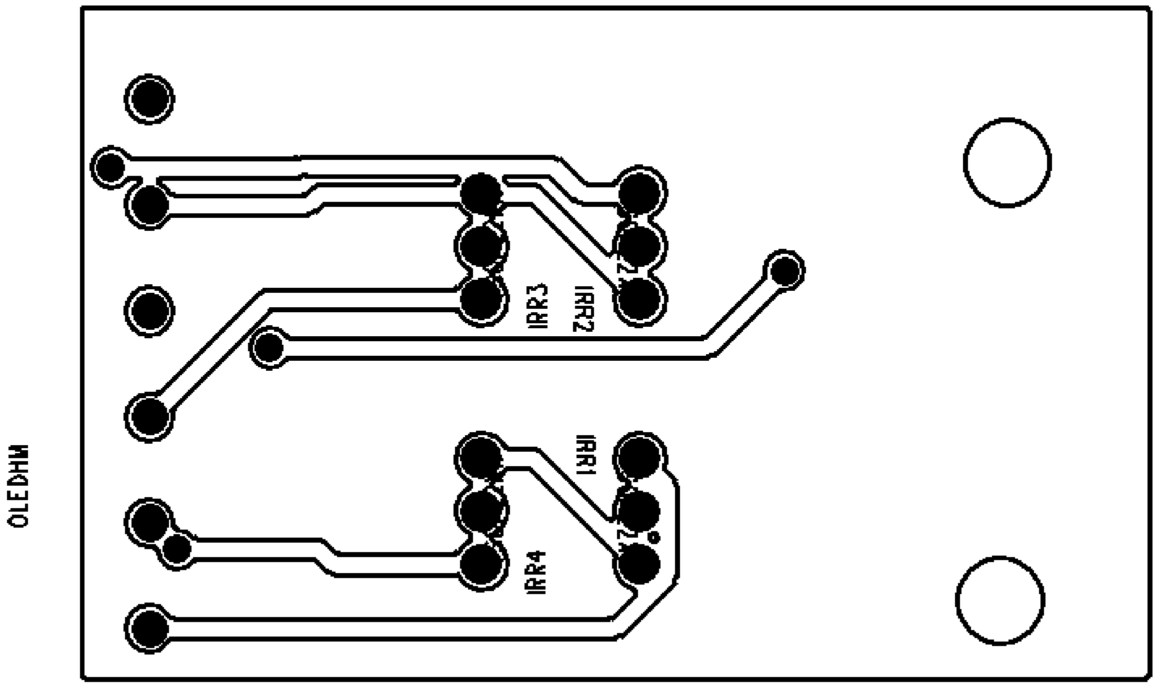

Figure D-3: IR Receiver Board Top Layer

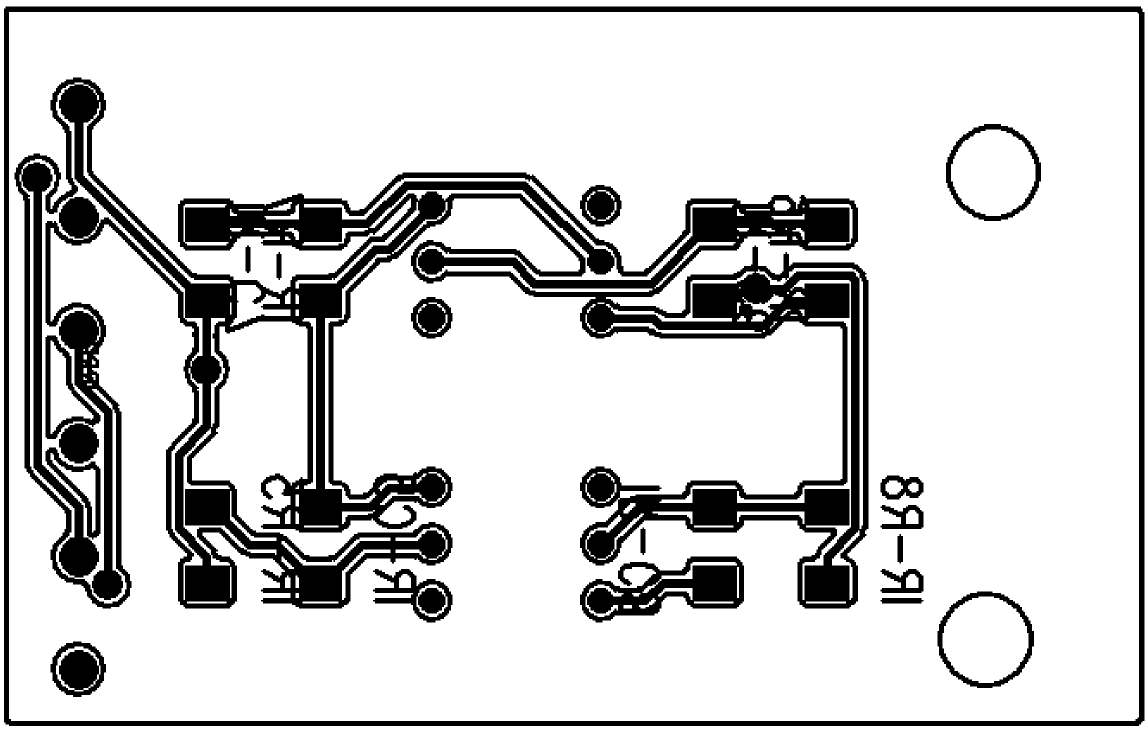

Figure D-4: IR Receiver Board Bottom Layer

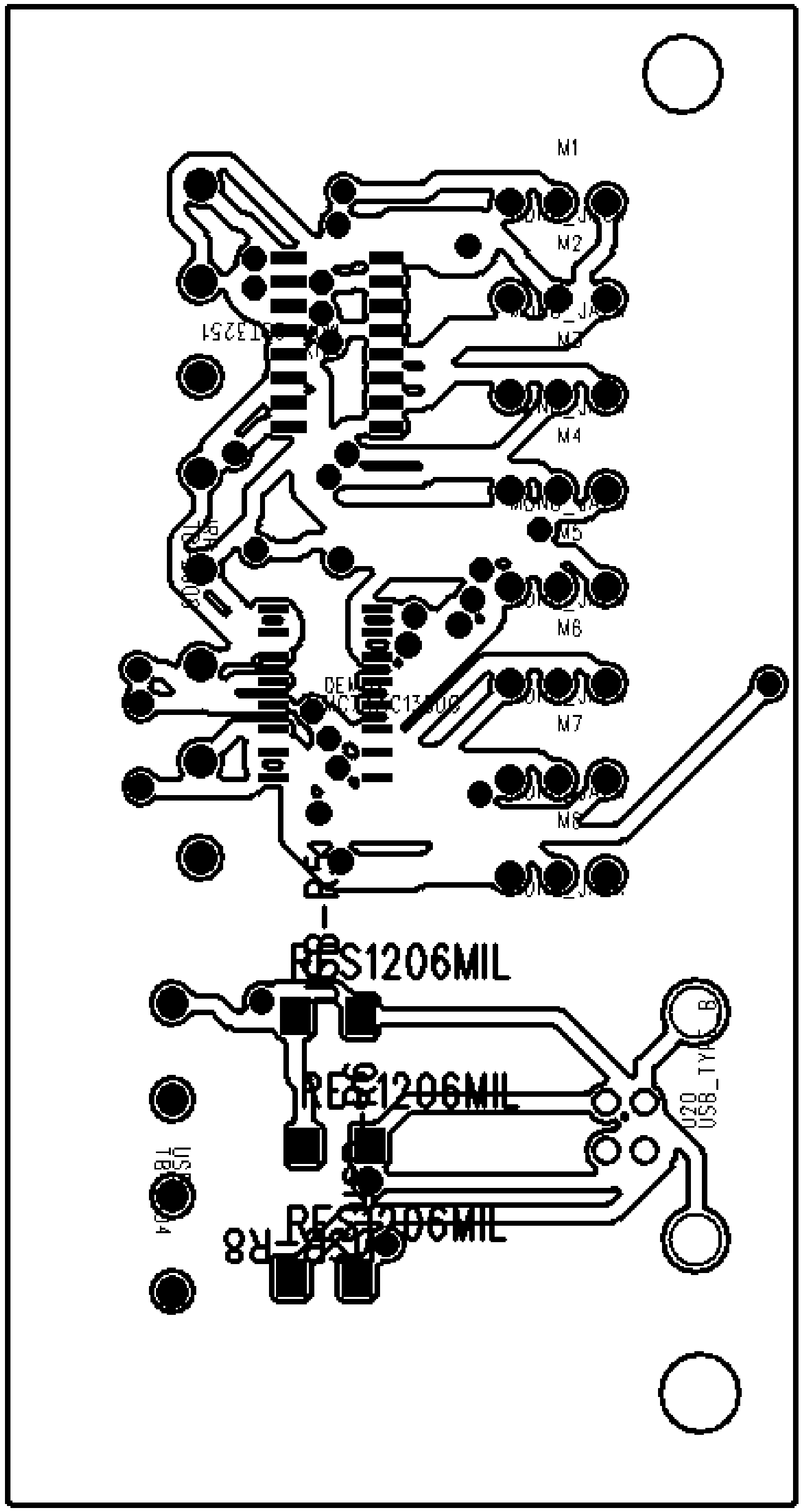

Figure D-3: IR Emitter Board Top Layer

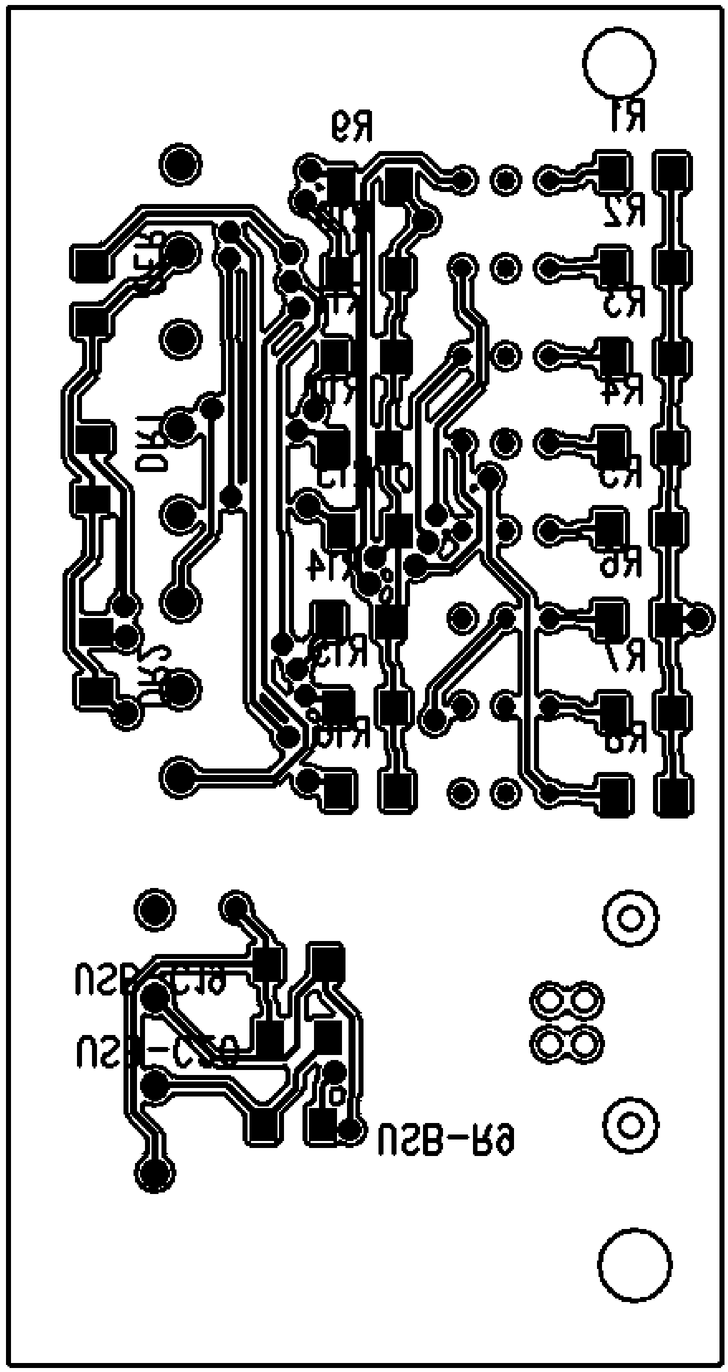

Figure D-3: IR Emitter Board Bottom Layer

Appendix E: Parts List Spreadsheet

Vendor	Manufacturer	Part Number	Description	Unit Cost	#	Total Cost
Roving Networks	Roving Networks	RN-121	WiFly RN-121 Wi-Fi module with Breakout	$69.00	1	$69.00
Mouser	OncQue	RBS070600	Tilt Sensor	$1.57	1	$1.57
DigiKey	Newhaven Displays	NHD-12232KZ-NSW-BB	Graphic LCD module	$19.87	1	$19.87
Digikey	Tyco Electronics	292304-1	CONN USB RECEPT R/A TYPE B 4POS	$1.71	1	$1.71
Digikey	NXP Semiconductors	LPC2378	ARM 32bit Microcontroller, 144LQFP	$12.98	1	$12.98
Mouser	Vishay	TSOP2238	IR receiver 38kHz	$1.05	1	$1.05
Mouser	Vishay	TSOP2240	IR receiver 40kHz	$1.05	1	$1.05
Mouser	Vishay	TSOP2256	IR receiver 56kHz	$1.05	1	$1.05
Mouser	Vishay	TSOP2236	IR receiver 36kHz	$1.05	1	$1.05
Mouser	ON Semiconductors	MC74AC138DG	2-6V 1-of-8 DEMUX	$0.86	1	$0.86
Digikey	NXP Semiconductors	568-3626-5-ND	IC 1-OF-8 FET MUX/DEMUX 16-SOIC	$0.60	1	$0.60
Summit Source	CableTronix	CTIR-1	Single IR Infrared Emitter 5' FT Cable 3.5mm	$3.95	4	$15.80
Digikey	CUI Inc.	CP-3502N-ND	CONN AUDIO JACK 3.5MM MONO	$0.83	8	$6.64
Digikey	ON Semiconductors	BC850CLT1G	TRANS NPN LP 100MA 45V SOT23	$0.40	2	$0.80
Digikey	Vishay	SI2323DS-T1-E3	MOSFET P-CH 20V 3.7A SOT23-3	$1.18	1	$1.18
Digikey	Linear Technology	LTC1174HVCS8-3.3#PBF	IC DC/DC CONV STP-DWN&INVRT8SOI(	$8.30	1	$8.30
Digikey	Linear Technology	LT1129CST-3.3#PBF-ND	LOW DROPOUT REGULATOR	$4.50	1	$4.50
Digikey	AVX Corporation	TPSD226K025R0100	CAP TANT LOESR 22UF 25V 10% SMD	$4.20	3	$12.60
Digikey	AVX Corporation	TPSD476K016R0150	CAP TANT LOWESR 47UF 16V 10% SMD	$3.90	2	$7.80
Digikey	Coiltronics/Div of Cooper/E	CTX50-4-R	INDUCTOR TOROID DUAL 50.18UH SMD	$5.68	1	$5.68
Digikey	Coiltronics/Div of Cooper/E	513-1198-1-ND	INDUCTOR TOROID 99.23UH DUAL SMD	$5.68	1	$5.68
Digikey	Fairchild Semiconductor	1N5818FSCT-ND	DIODE SCHOTTKY 30V 1A DO41	$0.54	3	$1.62
Mouser	Kobiconn	163-7620-E	DC Power Connectors PCB 2.1MM	$0.63	1	$0.63
Mouser	Xicon	412-109054	Plug-In AC Adapters 9VDC/500MA 2.1mm F	$10.05	1	$10.05
Future Electronics	Cherry Electronics	CRE22F4FBBNE	On-Off Quick Connect Power Rocker Switch	$0.88	1	$0.88
Digikey	ECS Inc	XC736 CT-ND	Crystal 12.000 MHz 22FF SMD	$0.52	1	$0.52
				TOTAL		$193.47

Appendix F: FMECA Worksheet

In the following tables, the word 'device' refers to the OMNiMOTE, while the word 'component' refers to something the OMNiMOTE is controlling, (i.e. TV, VCR, etc.)

Table 1: Power supply

Failure No.	Failure Mode	Possible Causes	Failure Effects	Method of Detection	Criticality	Remarks
A1	9V Rail = 0V	9V Power supply short, connector from 9V supply to main board failed	No power to any circuit components. Nothing works.	Observation.	High	If shorted, could cause a fire.
A2	9V Rail > 9V	9V Power supply has failed	Damage to regulators and LCD display	Observation	High	Overvoltage could cause fire.
A3	9V Rail < Lowest Tolerated Voltage	9V Power supply has failed, too much current draw from supply	Could possibly damage components or make them stop functioning	Observation	Medium	Probably won't cause a fire, but could damage components.
A4	5V Rail = 0V	Bad 5V regulator, short across 5V rail	5V components stop functioning (Wi-Fi module, Tilt sensor)	Observation	High	If shorted, could cause fire.

A5	5V Rail > 5V	5V power regulator bad	Could cause damage to 5V components (Wi-Fi module, Tilt Sensor)	Observation	High	Overvoltage could cause a fire.
A6	5V Rail < Lowest Tolerated Voltage	Bad 5V regulator, too much current draw from 5V regulator	Could damage 5V components or cause them to stop functioning	Observation	Medium	Probably won't cause a fire, but could damage components
A7	3.3V Rail = 0V	Bad 3.3V regulator, short across 3.3V rail	3.3V components stop functioning (Micro, LED's, mux/demux)	Observation	High	If shorted, could cause fire.
A8	3.3V Rail > 3.3V	3.3V power regulator bad	Could cause damage to 3.3V components	Observation	High	Overvoltage could cause a fire.
A9	3.3V Rail < Lowest Tolerated Voltage	Bad 3.3V regulator, too much current draw from 3.3V regulator	Could damage 3.3V components or cause them to stop functioning	Observation	Medium	Probably won't cause a fire, but could damage components

Table 2: LCD Display

Failure No.	Failure Mode	Possible Causes	Failure Effects	Method of Detection	Criticality	Remarks
B1	LCD Display doesn’t power on	Failure from LCD power supply, failed connector	No Display	Observation. Display has no power.	Low	Not a major failure, since the device can still function, but definitely makes it less convenient to use
B2	LCD Display is wrong/nonsense	Bad SPI communication from micro	User cannot view status of device	Observation. Display has no power.	Low	Similar to above, in that it doesn’t hurt the operation of the device
B3	LCD Display doesn’t rotate when device is turned.	Tilt sensor failed.	Display remains constantly in one orientation	Observation. Display has no power.	Low	May be hard to read, but definitely doesn’t hurt functionality.

Table 3: Wi-Fi Module

Failure No.	Failure Mode	Possible Causes	Failure Effects	Method of Detection	Criticality	Remarks
C1	Device can’t connect to network	Wi-Fi module failed, Wi-Fi reset pin stuck low	Device has been rendered null.	Observation. Device doesn’t appear on network from PC	Medium	This will cause the device to stop working altogether, but won’t cause any danger to user.
C2	Can’t send/receive data from device	UART pins have been damaged, Wi-Fi module has failed	Device can no longer operate	Observation. Device appears on network, but can’t connect via PC	Medium	Again, the device will stop working altogether.

Table 4: USB Interface

Failure No.	Failure Mode	Possible Causes	Failure Effects	Method of Detection	Criticality	Remarks
D1	Device can’t communicate with PC via USB	USB connector failure. Problem with connection to micro. Clock circuit failed.	Device can’t be setup using USB. Can’t initialize network information.	Observation. Doesn’t connect when plugged into PC.	Medium	This one is kind of between Medium and Low. It only matters if you change your network settings after the failure occurs.

Table 5: IR Emitters

Failure No.	Failure Mode	Possible Causes	Failure Effects	Method of Detection	Criticality	Remarks
E1	Can’t control IR component with single port.	IR emitter failure, IR resistor burned up	Can’t control any component on the port in which failure occurred.	Observation. Some components work and others don’t.	Medium	This might be considered low, since you can always use a different port. Also, in the case of emitter failure, you can just buy another emitter.
E2	Can’t control IR components on ANY port.	IR mux failure, micro PWM failure	Can’t control any ports at all.	Observation, device functions but no components can be controlled.	Medium	Obviously if this happens then the device is rendered useless.
E3	Can only control certain features of a component(i.e. power button works but not volume)	IR receiver filter is failing	Can’t mimic certain buttons on remote.	Observation. Only certain IR commands fail.	Medium	This could be low if it was a one time flaw in the signal. Can try to reprogram the button.

Table 6: IR Receivers

Failure No.	Failure Mode	Possible Causes	Failure Effects	Method of Detection	Criticality	Remarks
F1	No commands received when trying to program.	IR Receiver failure, IR capacitor or resistor failure.	Cannot program any commands	Observation. While trying to learn a command, nothing happens.	Medium.	Can no longer learn any commands, so could be critical if you get a new remote.
F2	Junk commands received when trying to program remote.	IR receiver failure, IR capacitor or resistor failure.	Cannot program any commands.	Observation. While trying to learn a command, device errors out.	Medium	Same as above.

Table 7: Microcontroller

Failure No.	Failure Mode	Possible Causes	Failure Effects	Method of Detection	Criticality	Remarks
G1	PWM output stuck high/low	Pin on micro damaged	Can't send IR commands.	Observation.	Medium	This causes entire device to become non-functional.
G2	Port select pins stuck high/low	Pins on micro damaged	Can only IR commands across one port.	Observation	Medium	Even though one port works, the rest can't be used.
G3	LCD UART output stuck high/low	UART3 pins on micro damaged	Can't display information on LCD display	Observation	Low	Doesn't cause device to become worthless, but is inconvenient.
G4	Wi-Fi UART output stuck high/low	UART1 pins on micro damaged	Can't communicate via Wi-Fi	Observation	Medium	All communication to device is gone.
G6	Constant low on reset pin	Micro reset pushbutton shorted	Micro is constantly being reset	Observation	Medium	If the micro is being reset, no functionality will be available.