Final Report for ECE 445, Senior Design, Fall 2018

TA: Nicholas Ratajczyk

12 December 2018

Project No. 12

Ye Dong Kim

John Peloquin

Sailesh Subashbabu

By

Vitals sensing for electronics-assisted therapy

**Abstract**

This project provides an upgrade for the existing Therapalz product used for Alzheimer’s patient care.  The design incorporates a group of sensors into the stuffed therapy animal that allow for non-invasive monitoring of the patient’s vitals signs and send the collected data over WiFi to a database with live display for the caretakers.  The final product is not complete; as will be discussed issues remain with the accuracy of two of the sensors and the implementation of the WiFi module, but engineering solutions have been found and the project was found to be a successful proof of concept for Therapalz.

Contents

[1. Introduction 1](#_Toc532414229)

[1.1 Block Diagram 2](#_Toc532414230)

[1.1.1 Sensor Module 2](#_Toc532414231)

[1.1.2 WiFi Module 3](#_Toc532414232)

[1.1.3 Control Module & Power Supply 3](#_Toc532414233)

[1.1.4 Computer Display 3](#_Toc532414234)

[2.1 Sensor Module 4](#_Toc532414235)

[2.1.1 IR Emitter/Detector and Force Sensor 4](#_Toc532414236)

[2.1.2 IR Thermometer 5](#_Toc532414237)

[2.1.3 Digital Thermometer 6](#_Toc532414238)

[2.2 Wifi Module 7](#_Toc532414239)

[2.3 Control Module 7](#_Toc532414240)

[2.4 Power Module 11](#_Toc532414241)

[2.5 Computer Display 11](#_Toc532414242)

[3.1 Sensor Module 12](#_Toc532414243)

[3.1.1 IR Emitter/Detector and Force Sensor 12](#_Toc532414244)

[3.1.2 IR Thermometer 12](#_Toc532414245)

[3.1.3 Digital Thermometer 12](#_Toc532414246)

[3.2 WiFi Module 13](#_Toc532414248)

[3.3 Microcontroller and Power Supply 13](#_Toc532414249)

[3.5 Display Module 15](#_Toc532414250)

[Figure 3.3 Final Webpage Highcharts Display 15](#_Toc532414252)

[4. Costs 16](#_Toc532414253)

[4.1 Labor 16](#_Toc532414254)

[4.2 Parts 16](#_Toc532414255)

[5. Conclusion 18](#_Toc532414256)

[5.1 Accomplishments 18](#_Toc532414257)

[5.2 Uncertainties and Future Work 19](#_Toc532414258)

[5.3 Ethical considerations 19](#_Toc532414259)

[Appendix A 21](#_Toc532414260)

[Requirement and Verification Table 21](#_Toc532414261)

# 1. Introduction

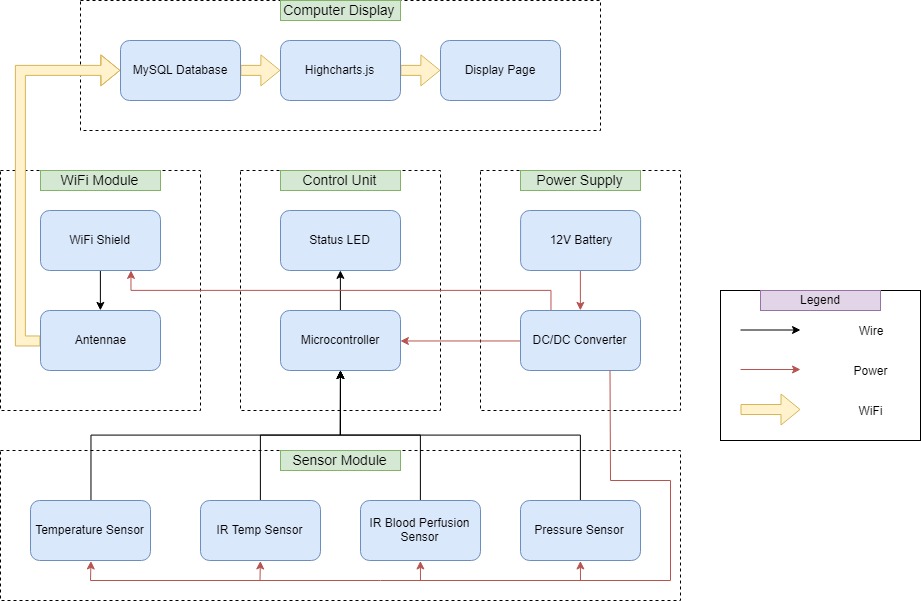
Alzheimer’s disease is one of the most devastating illnesses affecting people in America.  It is the 6th leading cause of death in the United States, with one in three senior citizens dying with Alzheimer’s or some form of dementia [1].  For the roughly 5.7 million people in America affected with the disease today there are some 16.1 million Americans providing care free of charge, an estimated 18.4 billion hours of care a year valued at over $232 billion [1].  Currently, every 65 seconds someone is diagnosed with the disease, and the number of those afflicted is expected to rise as high as 14 million by the year 2050. Therefore, the number of those providing this care would have to increase as well. Meanwhile, these men and women are dedicating this valuable time to those suffering while receiving very little outside support to make their jobs any easier.

Therapalz is a company that works to help remedy this situation by assisting with the care of Alzheimer’s patients.  They have developed a stuffed animal that uses haptic sensors to detect the touch of Alzheimer’s patients and respond with soothing noises and vibrations, replicating the feeling of a pet in order to provide comfort to the patient and make it easier for the caretakers to do their jobs.  However, one of the biggest issues the caretakers face is that they need to gather frequent data from the patients, such as heart rate or body temperature, that must be physically measured. This can confuse or upset the patients and cause stress to both parties.

The purpose of this project was to create a suite of sensors that could incorporate into the existing Therapalz product that would measure these same health indicators in a much less invasive fashion.  We chose sensors to measure the breathing rate, heart rate, and body temperature of patients, as those were some of the most frequently measured indicators. We then added WiFi and display modules to our design so as to allow the data gathered by the sensors to be sent remotely to an app that Therapalz would develop, later changed to be a database that could be accessed on a computer and would display the data in real time.  Our sensors, WiFi, and power modules would all be incorporated into the body of the stuffed animal itself, with the sensors placed at extremities to gather data and the PCB that controls the sensor suite in the main body with the WiFi and power modules.

The following chapters will describe the design process of each module of our project and the results of each individual section.  The project did not work perfectly, as not all of the sensors were as accurate as we required and the WiFi module did not work as intended during the prototype demonstration.  However, the main conclusion was that the project served as a good proof of concept to Therapalz and that engineering solutions for all of the issues encountered exist and could be implemented with more time.

## 1.1 Block Diagram



**Figure 1   Block Diagram of Design Modules**

This figure outlines each module of our overall design and the individual components that comprise them.  Wiring, wireless, and power connections are included to show how the components relate to each other and will be elaborated on for each module.

### 1.1.1 Sensor Module

The sensor module comprises the bulk of our project.  The pressure sensor is used in tandem with the IR (infrared) blood perfusion sensor to monitor the heart rate of the patients, with the pressure sensor serving as a trigger for when the blood perfusion sensor should gather data.  The IR temperature sensor is to be used to measure the body temperature of the patient, while the purpose of the digital temperature sensor is to measure their breathing rate. All of these feed their data into the control module.  Both thermometers needed to be able to report temperatures accurate within ±1 ˚C and output at least one reading per second, while being placed in locations that ensured they would get meaningful readings in a 30-minute session of use by the patient.  The IR blood perfusion sensor was required to output 10 readings per second when in use, and measure heart rates accurate to ±10 BPM, changed from ±20 BPM in the original design. The force sensor requirement was simply to withstand the 14.5N maximum pressure.

### 1.1.2 WiFi Module

The WiFi module serves as the bridge between the hardware side of our project and the backend software.  Data is sent from the control module to the WiFi chip we selected, which connects to the local network of where the Therapalz product is being used and transfers the data to the display module.  The requirement was that the module successfully perform its task without any packet loss or disconnections.

### 1.1.3 Control Module & Power Supply

The power module supplies power from a battery to the microcontroller in the control module.  This microcontroller is then used to convert the supplied power into the different power requirements for each sensor to power them on.  The sensor data is then received by the microcontroller in its raw form and held until the WiFi module gathers it. A status LED is included to help determine when the system is operating in the ON state.  The power supply needed to provide 7.4V ±0.2V for the microcontroller and allow the entire design to operate for 1 hour in the ON state and 10 hours in the idle state without needing a recharge. The microcontroller module had to be able to supply 3.3±0.2V and 5.0±0.2V with respect to its internal ground and was updated from the original design to include a requirement of supporting integration of all four sensors at once.

### 1.1.4 Computer Display

The computer display is responsible for the presentation of the data our project is designed to gather.  It had to include an algorithm to filter the data gathered from the sensors into the database, and a display page that showed the data in real time updating every second.  It was also required that the display module database must be able to accept 1 multi-column input per second and log everything on the fly with a maximum 2 second latency.

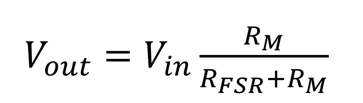
**2 Design**

# 2.1 Sensor Module

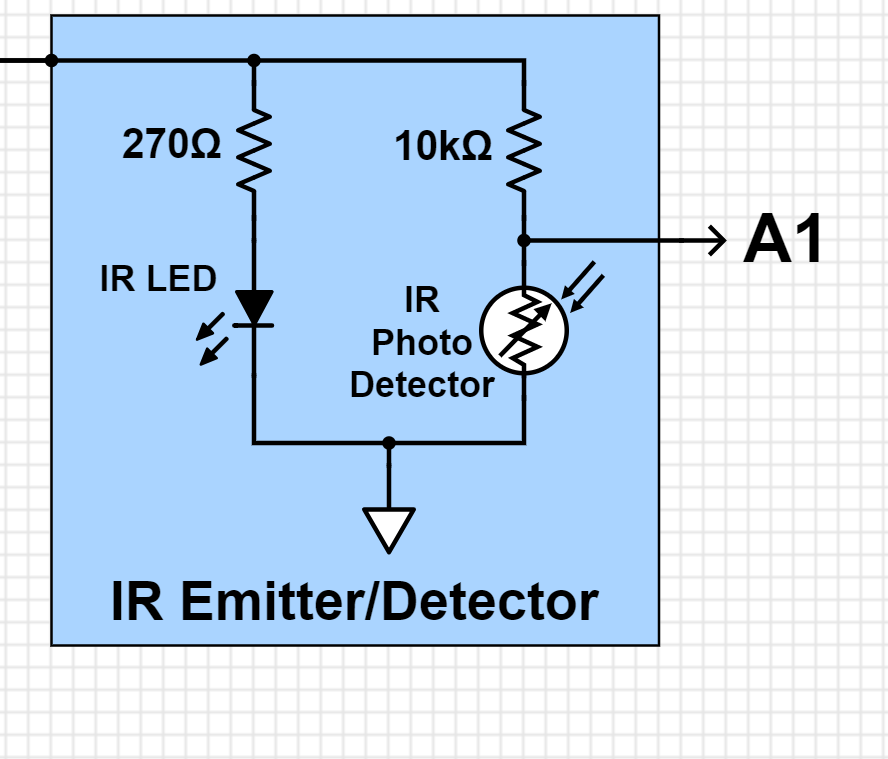
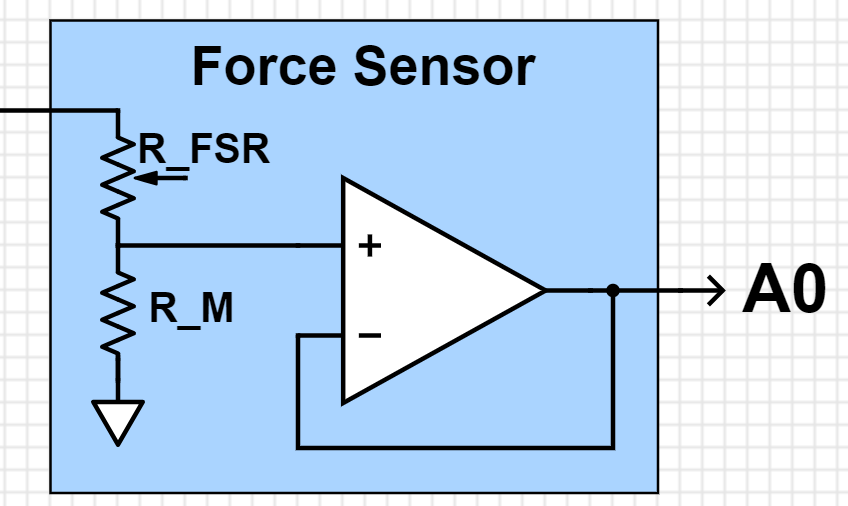
### 2.1.1 IR Emitter/Detector and Force Sensor

The infrared emitter and detector is used to measure heart rate by detecting the change in blood flow in the patient’s fingers. Because of cost and size constraints, we were looking for an infrared sensor that had a high enough accuracy to detect slight changes in infrared intensity and that was small enough to be placed inside the paw of the product. The SEN00241 that we chose is an IR emitter and detector pair that hit both of these specifications, while also being very well documented to be able to perform with a low supply voltage, high reliability, and being able to do so without heating up, which would be a safety hazard.

As blood is pumped through veins, the finger becomes slightly opaque, resulting in a change in reflected infrared intensity. The photodetector will respond by changing the voltage at its output, A1, shown in **Figure 2.1.1**. Using a peak-finding algorithm, the periodicity of the peaks can be measured and translated into heart beats per minute, BPM. Considering the amount of power that would be wasted if it were continuously taking heart beat measurements, we used a force sensor to accompany the IR emitter and detector pair. The force sensor is triggered when it detects a slight amount of force, which then activates the heart rate measurement sensors. It works by acting as a potentiometer in a voltage divider shown in **Figure 2.1.2**; the greater the force it detects, the lower its resistance is. Using a simple voltage divider equation in Equation 1, a greater force applied to the force sensor results in a larger output voltage.

       [1]

We had two perfboards side-by-side with the force sensor on one side and the infrared emitter and detector closely placed together on the opposite side. Both of these were shaped in order to be able to fit in the pet’s paw. We used two perfboards together so the force sensor would be more easily triggered inside the paw of the pet while also providing a solid foundation for the sensors to be stuck onto to take better measurements. One constraint to this, however, is that the patient needs to keep their finger on the infrared emitter and detector without moving too much. Otherwise, the sensors would have a more difficult time differentiating between noise and actual heart beats.

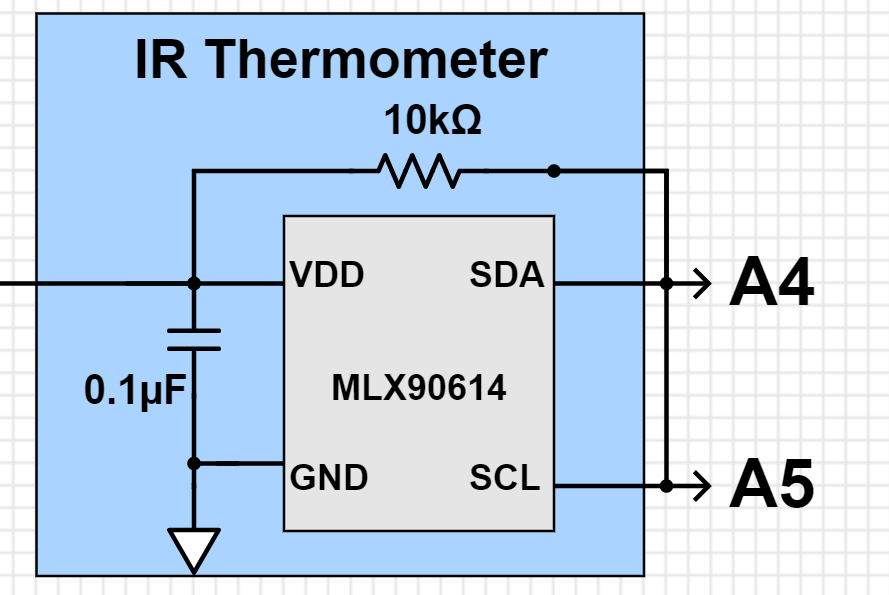
**Figure 2.1.1 Schematic of IR Emitter/Detector            Figure 2.1.2 Schematic of Force Sensor**

### 2.1.2 IR Thermometer

Our design called for a way to measure the body temperature of a patient without having to come into physical contact with them.  Through our research we determined the best way to do so was with an IR (infrared) thermometer. We selected the MLX90614 made by Melexis because their website stated that their product is often used for medical practices, making it a good fit for our design requirements of high accuracy and frequent measurements.

This sensor uses an infrared-sensitive thermopile detector chip to detect the ambient temperature around the sensor, sending that value as output while storing it for the next step.  The infrared beam of the sensor is used in conjunction with a pre-programmed emissivity coefficient that compares how well a detected object emits infrared radiation in comparison to a theoretical blackbody that Melexis approximates. [IR THERM]  Because all of this is done within the sensor’s single “can”, we were able to keep our circuit fairly simple so as to fit into the body of the Therapalz product.

As shown in **Figure 2.1.3**, we needed only a 10kOhm resistor and a 0.1uF capacitor to prevent damage to the circuit and filter out unwanted frequencies.  The sensor uses two analog output pins for sending data writing and clock management to determine when data is collected and sent to the microcontroller.  Our design originally had the sensor placed in the nose, as patients would often look into the stuffed animal’s eyes when playing with it. Due to the stitching on the animal we could not fit the perf board we placed the circuit on into the nasal cavity, so we placed it on the outside of the nose.  We also considered placement in the eye itself, but that would require a recalculation of the emissivity coefficient that will be discussed more in the verification chapter.

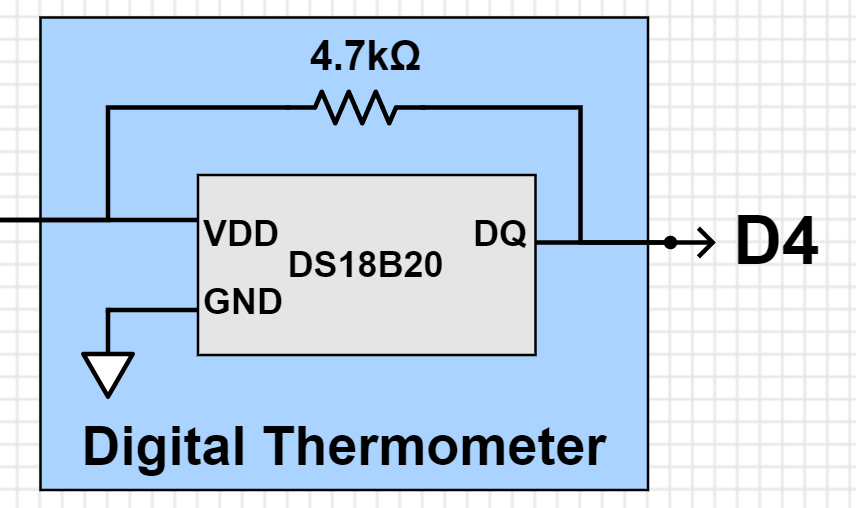


**Figure 2.1.3 IR Thermometer Schematic**

### 2.1.3 Digital Thermometer

In order to measure breathing rate, we required a thermometer that could detect the changes in temperature around a patient’s mouth. Thus, the requirement for accuracy wasn’t very high but the thermometer did have to be sensitive and small enough to be placed in the head of the product. We initially designed it to be placed near the neck area of the pet but later revised it to be placed at the top of the head. This way, the sensor could be closer to the patient’s face while still being able to be disguised. As the digital thermometer picks up changes in temperature, a peak-finding algorithm will once again find the periodicity of peaks corresponding to when the patient breathes out.

We selected the DS18B20 “One Wire Thermometer” from Maxim Integrated because of its reliability and the fact that it only required one port pin for communication with the microcontroller, as shown as “DQ” in **Figure 2.1.4**. The DS18B20’s supply voltage is the microcontroller’s 3.3V output.



**Figure 2.1.4 Digital Thermometer Schematic**

## 2.2 Wifi Module

We would require a WiFi model that was low cost, and could transmit data reliably. While shopping for potential products, we found the ESP8266, which was a well-documented, affordable standalone chip with WiFi capabilities that had been used in many projects by Arduino enthusiasts. We specifically looked for the capability to publish data to a web page or database, and learned that the module could use the HTTP protocol, allowing this type of communication. Given that this was a standalone chip, we would require an FTDI USB-to-serial converter to flash it as well.

## 2.3 Control Module

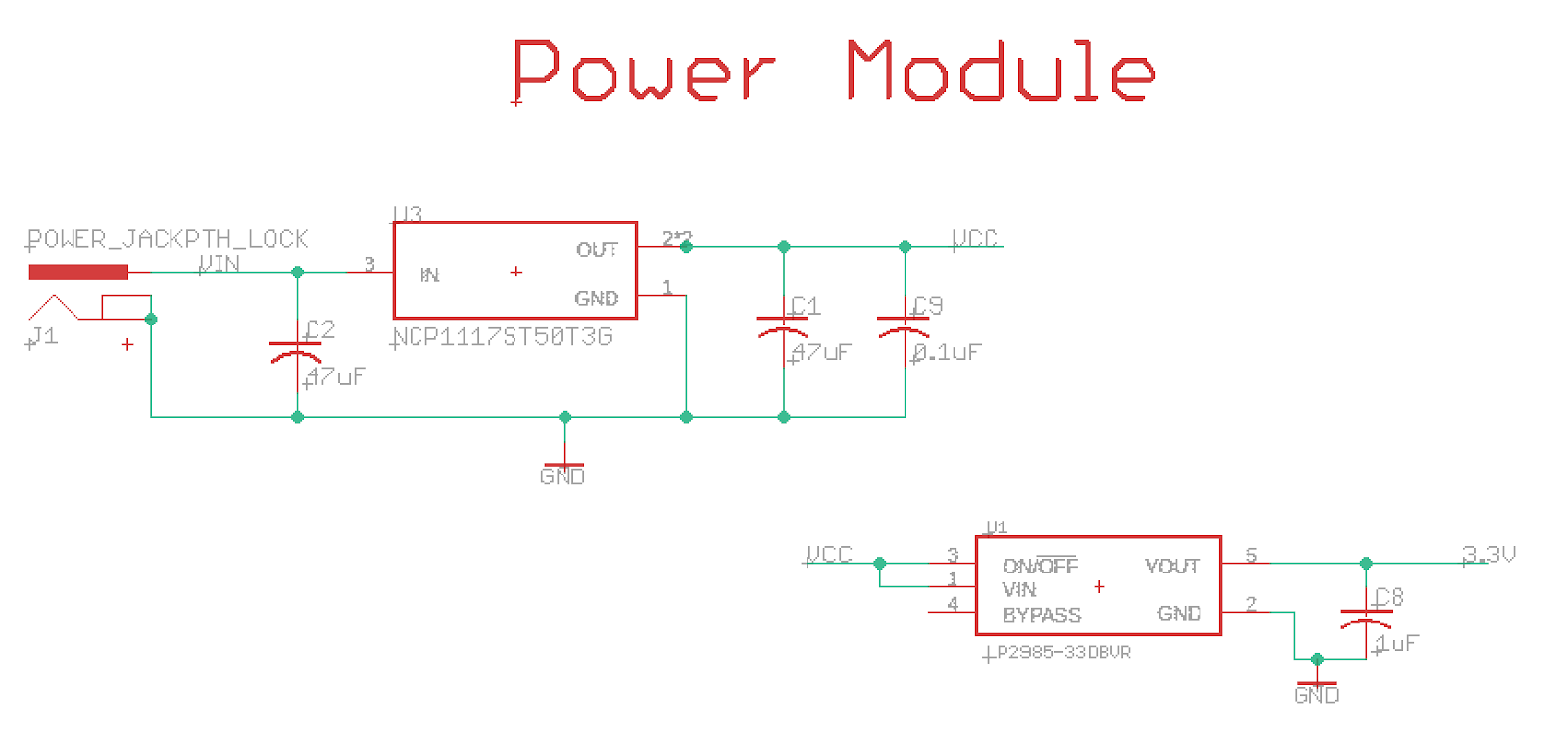
We required a microcontroller that can output around 3.3V and 5V for our sensors. It also had to be able to simultaneously process all the date from the sensors and smoothly transfer the data to a compatible wifi shield to transmit the data over WiFi. Thus, we looked at the Arduino products because of its open-source hardware and available board schematics. We settled for the ATmega328P microcontroller (used in the Arduino Nano, Uno, and Sparkfun Redboard products) because of its low price, suitable performance, and memory space for our small scale product. An alternative microcontroller that went into consideration during the design stage was the ATmega2560 used for in the Arduino Mega board but we decided an ATmega328P would suffice.

We then designed an ATmega328P based PCB microcontroller with only the functionalities we needed in order to also cut down on board space. The control module can be broken down into three components; the microcontroller module, power module, and input/output module. These are shown below in **Figures 2.3.1**, **2.3.2**, and **2.3.3**, respectively.

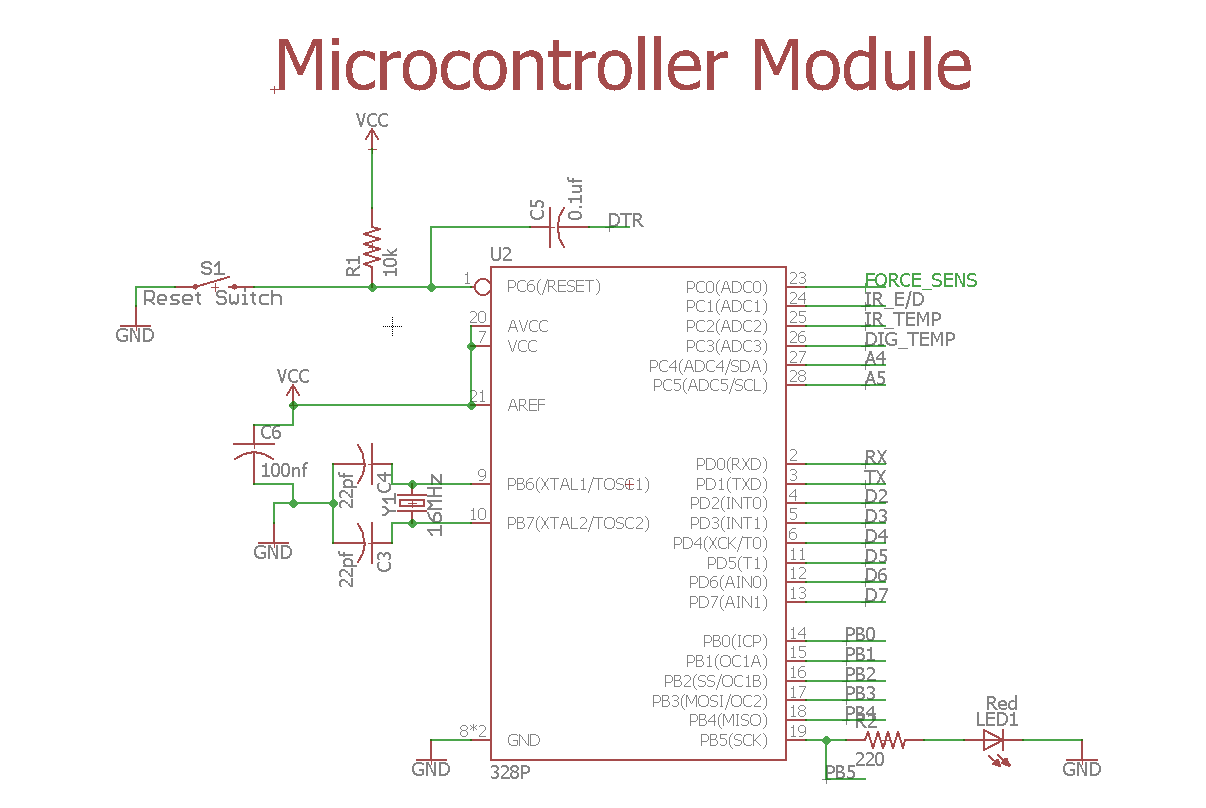
In the power module, we have two voltage regulators. The first one takes in 7.4V input voltage and outputs a regulated 5V output. The microcontroller operates at 5V so this becomes “VCC”. The 5V is then fed in to another voltage regulator to output 3.3V which is used for some of the sensors. Both circuits have a combination of bypass capacitors to help regulate the voltages. Both voltage regulators are SOT-type SMD packages. The input/output module was used for the sake of convenience during the debugging process of the PCB such as using them as voltage test points. Female pin headers are also soldered on to allow flexibility in connecting different sensors or devices to the microcontroller’s analog/digital pins as well as to its 5.0V and 3.3V power supply.

The microcontroller module has a built in LED to help during the debugging process to make sure the sensors are correctly communicating with the microcontroller. In **Figure 2.3.2**, you can see how the analog, digital, and TX/RX pins are utilized to connect all the sensors and WiFi module. The sensors are provided their supply voltages from the microcontroller and are given designated output pins that correspond to the input pins on the microcontroller. Shown in **Figure 2.3.4**, the microcontroller PCB has decoupling capacitors on-board but additional bypass capacitors at each supply voltage lines further reduce voltage ripples and noise. This is important, as sensor measurements are sensitive to the device’s supply voltage.

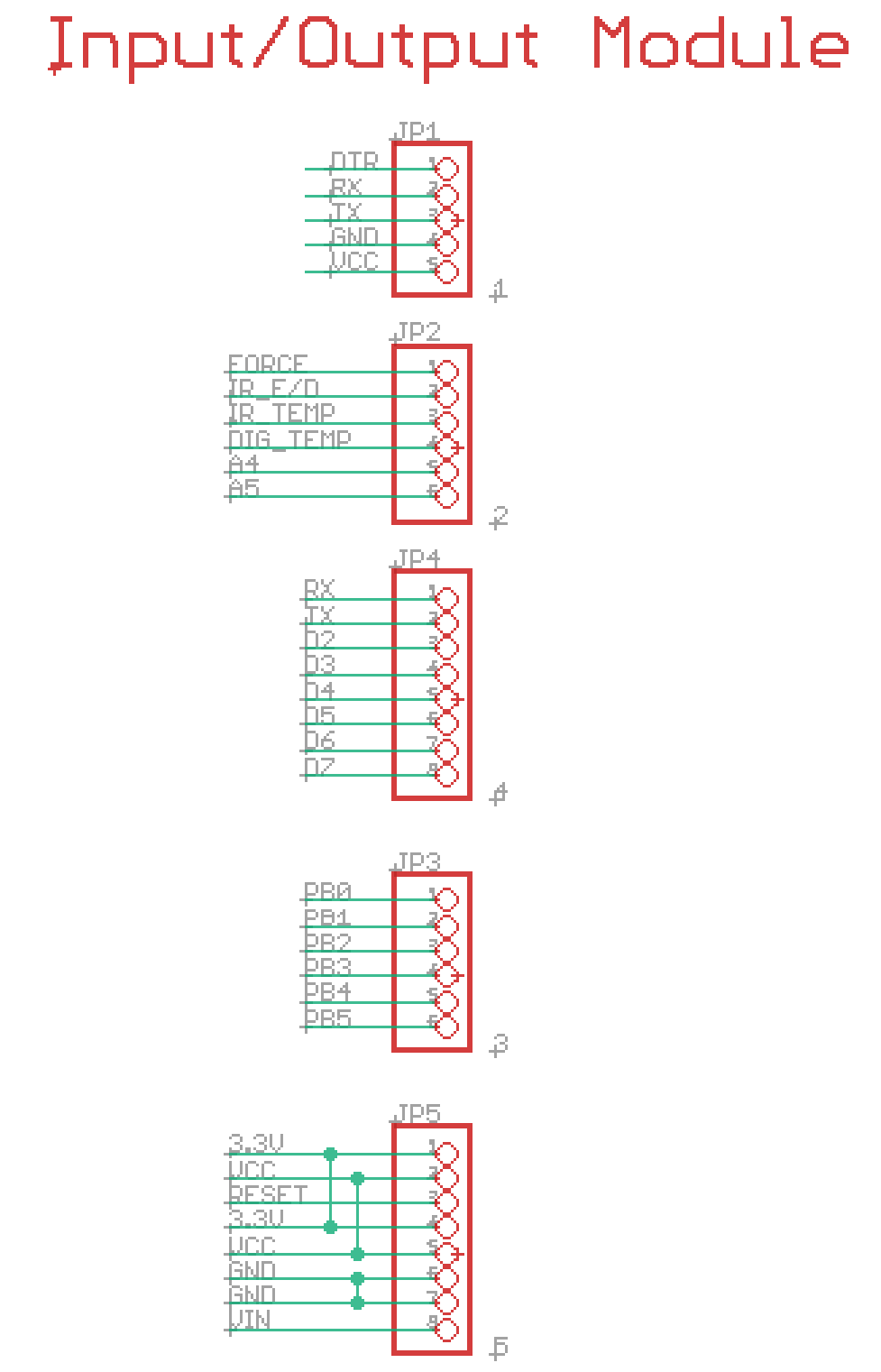
The final result of these schematics is the PCB shown in **Figure 2.3.5**. The power module is located on the left side, microcontroller module on the right side, and input/output headers scattered around the board. Headers JP5 corresponding to the output supply voltages are placed close to the power management while analog and digital pin headers JP2 and JP4 are located closer to the microcontroller side.



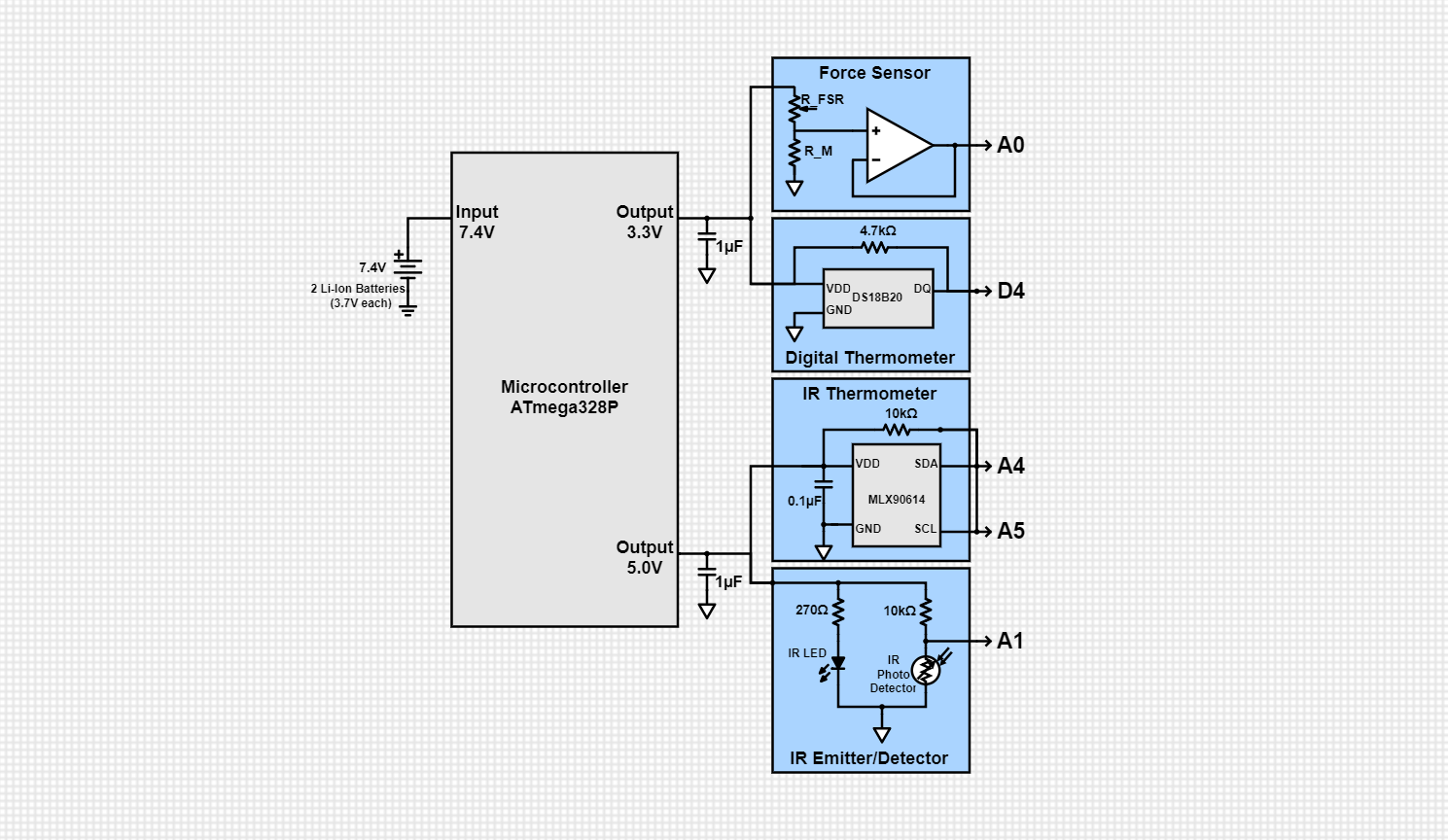
**Figure 2.3.1 Power Module Schematic**



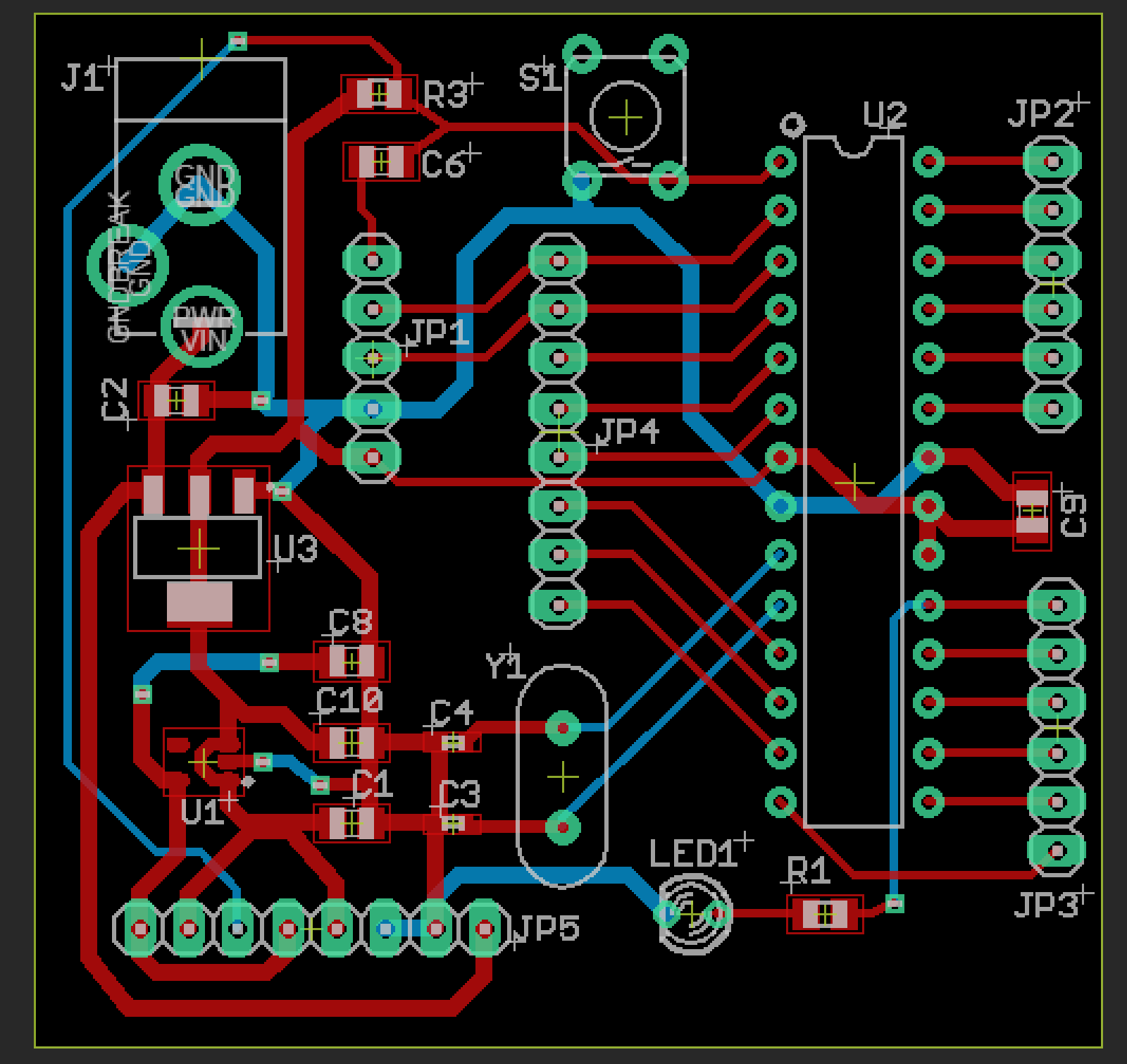
**Figure 2.3.2 Microcontroller Module Schematic**



**Figure 2.3.3 Input/Output Module Schematic**



**Figure 2.3.4 Schematic of Microcontroller and Sensors**



**Figure 2.3.5 Final PCB Design Layout**

## 2.4 Power Module

During the design stage, we weren’t sure how much power the entire system would draw but we did have specifications on how much power should provide. Therapalz has their own integrated product they wish to use for the final power/charging system, so our power source was required for our testing and charging purposes. We needed batteries that would be placed inside a battery case that would provide the microcontroller enough voltage and charge. Moreover, the batteries had to be rechargeable and not too chunky as it would also be placed inside the product. We expected the patients to be using the pets quite frequently, so we need enough battery capacity to last the pet at least 1 hour at its ON state and at least 10 hours in its IDLE state off fully charged batteries.

The ON state is activated when the maximum amount of current is drawn from the batteries. This is the state when all the sensors are drawing power to send data to the microcontroller and WiFi module, including the power drawn from the WiFi module to transmit the data over to the database. The IDLE state refers to when the sensors are not necessarily all drawing power from the batteries. This can occur if the sensors are taking junk data; e.g. the infrared thermometer filters out temperature readings that are outside the human body temperature range, or if the infrared emitter and detector pair aren’t being activated because the force sensor is not being triggered.

Thus, we settled for two Li-Ion rechargeable batteries in series - both 18650 battery type with 3000mAh and 3.7 V with a combined voltage of 7.4V - and a battery charger that converts grid power to a regulated 4.2 V, which then charges up 2 Li-ion batteries. We chose 18650 batteries because they had the most charge for their size and it was a better alternative than five AA batteries because that would have taken up more space. We chose a 3000mAh battery because we estimated it would suffice for our project. 3.7V batteries are characteristic for Li-Ion batteries and it worked out that we only needed two in series to provide enough input voltage for the microcontroller to run.

## 2.5 Computer Display

For the display, we wished to view the sensor data in real time. This required that we communicate between the microcontroller and computer. We also knew that the sensor data would need to be filtered through some external method, as filtering within the microcontroller could decrease sampling drastically. Given these conditions, we felt the best method to receive data was a MySQL Database hosted through Xampp, as there could be a table for both the “raw” and “filtered” data respectively. This raw data could be filtered via a Python script, and inserted into the “filtered” table.

Another upside to using Xampp was the fact that the platform easily integrates and incorporates web scripts. The actual charting method was determined by searching for a Javascript module, as HTML scripting was unfamiliar territory for us. We stumbled upon Highcharts.js, and determined that the interactive plots and simplistic design suited a user that may not have an abundance of technical knowledge.

**3. Design Verification**

## 3.1 Sensor Module

## 3.1.1 IR Emitter/Detector and Force Sensor

To test if the heart rate sensor works, we uploaded a simple code that begins taking heart rate measurements when the force sensor delivers a voltage greater than a certain threshold at its output. After optimizing the threshold value such that the readings would begin after some slight pressure on the force sensor, we began seeing a series of values (0-255) that corresponded to voltages (0-5V) at the microcontroller’s heart rate input pin. We placed an LED that pulsed every time the voltage passed a certain threshold that indicated when a heartbeat was detected. We saw a periodic pulsing of the LED that confirmed that the detector was able to measure heart rate and only when the force sensor was triggered on, as designed.

## 3.1.2 IR Thermometer

The testing of the IR Thermometer also required a basic Arduino script that allowed us to determine both the ambient temperature and the object temperature that the sensor was detecting.  When we held the prototype with the IR Thermometer embedded in the nose area it did distinguish between the tester and the objects behind it, and produced the required one reading per second of use.  Unfortunately, while the ambient temperature around the sensor produced values that indicated a correct reading of room temperature (between 70-74˚F) the values given for the object, in this case the human body, peaked between 80-85˚F, which would indicate the subject was dead.  The values for the room temperature allowed us to assume the sensor was in good working order, so our conclusion was that either the emissivity coefficient of the sensor was not calibrated to be used on the human body or the sensor was simply not sensitive enough for our designed purpose.  We did verify that the location of the sensor was good, as it easily distinguished between the user and background objects, meaning the only missed verification was temperature accuracy.

## 3.1.3 Digital Thermometer

For the verification of the digital thermometer requirements we once again wrote an Arduino script that would allow us to find the detected temperature values.  In this case, all of the listed requirements were verified, but another issue arose. The sensor produced the required one reading per second, verified by watching a live update on the serial terminal of the Arduino and again with the finished product by looking at the database where we timestamped all of the data.  The original location was determined insufficient to detect consistent breathing from the patients so we moved it to the head, which was verified by our Therapalz contact to be a good location. Finally, comparing the detected room temperature with our known value we saw that the sensor was accurate enough at room temperature.  The complication we ran into that could have been an extra requirement was that the sensor was not sensitive enough to detect changes in temperature produced by breathing unless an extended exhalation was used or the user was extremely close to the sensor, neither of which would be ideal situations for field application of our design. All sensor data can be seen in **Figure 3.1**.

## https://lh5.googleusercontent.com/rrD7Oa-_Xj2aN5xrywOlL2aS00o8_tlVafeDtUrshYHXhbxNLT9Eb-qEG4J_X-OD9wvJvPvIBNQxKcJFEUf3rJHZbdf8r9Kgvqn1c8ehCNBwgVRyMNMGmj9RK0vnTfEH0O1ArAR6

**Figure 3.1 Live Sensor Data Timestamped within Database**

## 3.2 WiFi Module

In order to test communication between the ESP8266 and the ATMega328P, SoftwareSerial protocol was used to echo a message to the ESP8266, which would decipher that message, and send the deciphered message or acknowledgement back to the ATMega328P to be viewed in Serial. Once communication had been established, the Wifi connection was tested by using the base “WIFI.connected()” function, which echod an error code of 3, indicating a secure connection. Finally, the latency of connection and POST timing for data being sent to the database was monitored by timestamping the data as it exited the ESP and as it entered the database, indicating that the R & V had been met.

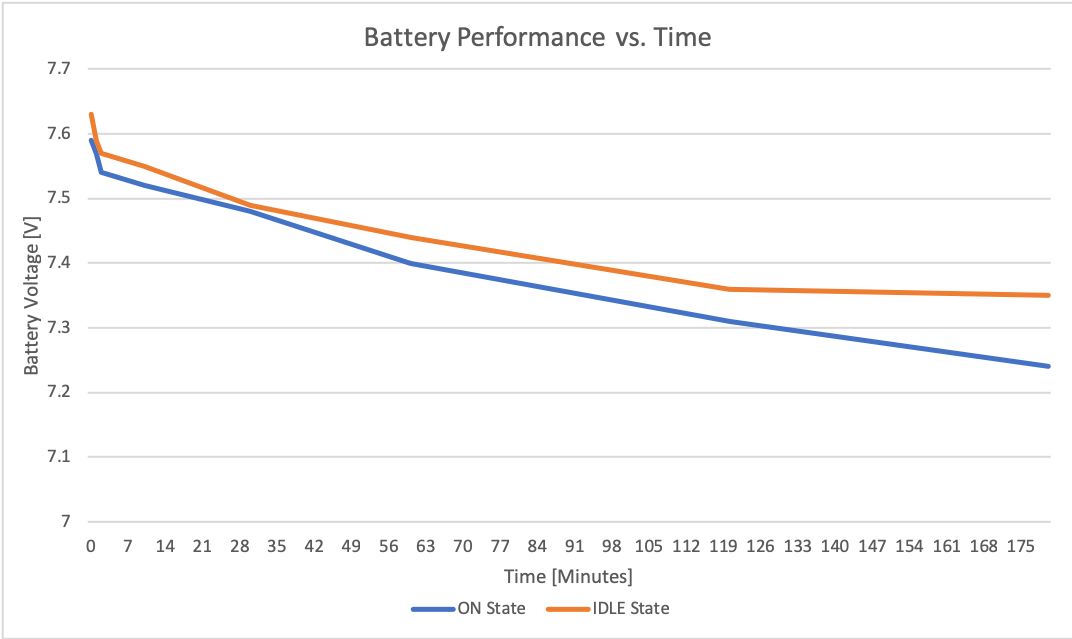
## 3.3 Microcontroller and Power Supply

To test the microcontroller’s output voltages, we used a power supply at 7.4 V that became the input voltage for the microcontroller. Using a multimeter, we measured a minimum of 4.997 V at the 5.0 V output and a minimum of 3.134 V at the 3.3 V output. All sensors continued to work at these minimum voltages. The microcontroller successfully processed the data for all sensors and was able to do so very quickly. However, when the WiFi module was introduced, the microcontroller slowed down significantly because of the delay times the WiFi module demanded.

Below 7.00 V, the microcontroller would not reliably output 3.3 V and 5.0 V, so we measured the time it took for the batteries to last from their fully charged battery voltage to 7.00 V at both the ON state and IDLE state. Table 1 presents the battery voltage measured over time and Figure 3.X shows the performance over time.

**Table 1 Battery Voltage over Time**

|  |  |  |
| --- | --- | --- |
| Time Taken [Hr:Min] | Battery Voltage, ON [V] | Battery Voltage, IDLE [V] |
| 0:00 | 7.59 | 7.63 |
| 0:01 | 7.57 | 7.59 |
| 0:02 | 7.54 | 7.57 |
| 0:10 | 7.52 | 7.55 |
| 0:30 | 7.48 | 7.49 |
| 1:00 | 7.40 | 7.44 |
| 2:00 | 7.31 | 7.36 |
| 3:00 | 7.24 | 7.35 |



**Figure 3.2 Battery performance over time**

## 3.5 Display Module

To verify the charts were working as intended, live data was fed into the “raw” table and filtered into the “filtered” table. The Highcharts module was updated every second to poll new data, and given the timestamp, we were able to verify that the latency was sub 2 seconds. The final viewing page for nurses and caretakers alike can be seen in **Figure 3.3**. We went one step further in verifying the time it took to filter the data from “raw” to “filtered” by finding the time difference between the initial SELECT call to grab the raw data, and the final INSERT call to send data to the “filtered” table. In all cases, this timing was less than 30 ms, so it wasn’t a bottleneck.

# https://lh4.googleusercontent.com/BkgR1rV3YfRWMlxscsiHhvI5mfebGkPVuxVupUty0DoAPx0mLAm3iVXLAirCb2alKe5vZsPjY_khed_HLnQjIQYCywHRjDTGfE9mSyNz_9e6cQGXjPx1UYKpnkMBMrCReYKm7XdM

# Figure 3.3 Final Webpage Highcharts Display

# 4. Costs

## 4.1 Labor

Our development costs were fixed at $40/hour and 6 hours/week for three people as a labor estimate.  We only consider the manufacturing of our portion of the Therapalz product as the existing portion is assembled elsewhere, instead counting the stuffed animal and motor as a fixed cost per unit at $50.  Only an estimated 80% of our product is considered in manufacturing costs over the scope of this course (16 weeks), neglecting the setup of the computer display and WiFi connection at whatever location the Therapalz product is being used at

3\*$40hour\*6 hourswks\*16 wks0.8\*2.5 = $36,000

This puts our design at an estimated $36k of labor pay for a full-time engineer.

## 4.2 Parts

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table 2   Parts Costs** | | | | |
| **Part** | **Manufacturer** | **Retail Cost ($)** | **Bulk Purchase Cost ($)** | **Actual Cost ($)** |
| ESP8266 Wifi Module | Mackey Tech | 1.75 | 1.75 | 1.75 |
| Infrared Emitter/Detector SEN 00241 | Sparkfun | 1.95 | 1.76 | 1.95 |
| FSR400 force sensing resistor | Interlink | 12.54 | 6.16 | 12.54 |
| 2 10uF capacitors, 2 1uF caps, 3 10kOhm resistors, 1 220Ohm resistor, 1 LED, 100nF cap, 1 4.7kOhm | Various, distributed by Digikey | 5.00 (estimated) | 2.00 (estimated) | Free |
| MLX90614 IR Thermometer | Melexis Technologies | 19.95 | 17.76 | 19.95 |
| PCBs | PCBway | 21.00 | 21.00 | 10.00 |
| Li-Ion Battery (x2) | EBL | 9.00 | 9.00 | 9.00 |
| Li-Ion Battery Charger | EBL | 10.00 | 10.00 | 10.00 |
| Therapalz Base Product | Therapalz | 50.00 | 12.00 | Free |
| Atmega328P | Microchip Technology | 2.14 | 1.78 | 2.14 |
| DS18B20 Digital Thermometer | Maxim Integrated | 2.78 | 2.72 | 2.78 |
| **Total** |  | **136.11** | **85.93** | **81.11** |

# 5. Conclusion

## 5.1 Accomplishments



**Figure 5.1 Final Product**

To summarize, despite some setbacks our design served as a good proof-of-concept for the product Therapalz hoped to see.  Two of our sensors are not accurate enough for our purposes, and the WiFi module did not work as intended during the demonstration phase but we believe relatively simple engineering solutions exist for all of these uncertainties.  Ultimately the design is still viable and Therapalz can continue with the project after implementing our proposed solutions to improve the lives of Alzheimer’s patients and their caretakers.

## 5.2 Uncertainties and Future Work

As described above and in the verification chapter both of our thermometers did not work as we had hoped.  The digital thermometer has the simplest proposed solution in that a more sensitive model could be purchased and implemented in the same manner our design used.  We chose the DS18B20 for its low cost as we operated on a thin budget, but for future testing Maxim makes higher quality models that could be looked into. The IR thermometer was not able to accurately detect a human body temperature unless the sensor was placed in the mouth or the ear as a normal thermometer would be, but there are two possible solutions for this.  The simpler one is that Melexis makes a medical grade model upon custom order that is meant specifically for human body use that was outside of our budget range. If budget constraints remain a concern, then it is possible to recalculate the emissivity coefficient used to compare the object temperature to the ambient temperature and match it more closely to a human body than the theoretical blackbody that was used, without the need for recalibrating the sensor itself. [2]

We are still uncertain as to why exactly the ESP8266 chip did not work as intended during the demonstration, but considering that it performed exactly as expected using the Arduino Uno in pre-testing and the only issue arose when we used our PCB we propose that it was a simple power issue.  Further testing would need to be done to confirm this, but assuming that is the case the engineering solution would be to supply more power to that module.

The next steps of this project would be to implement our sensors into a Therapalz product in the pre-production stage, as our sensors could not be placed fully inside the animal as intended due to the stitching getting in the way.  Repeating our same tests on this new prototype would allow any future engineers working on the project to determine how inaccurate the thermometers are before implementing the solutions outlined above.

## 5.3 Ethical considerations

Any future work on this project would have two primary ethical concerns to keep in mind. In terms of safety, as this is an electrical device contained within a stuffed animal there is a need to ensure that any modifications maintain the same safety standards as the existing electrical components of the Therapalz.  All components of this design would need to be thoroughly tested in order to confirm that no one will come to harm by using this product. This will serve to prevent the possibility of the product catching fire and harming the patients or caretakers.

As the sensors will serve to gather personal medical information on the patients, potentially without their direct knowledge given their condition, any future designers must be sure to maintain their privacy to the standards of their caregivers.  This will be in keeping with the first tenant of the IEEE code of ethics, “to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, and to disclose promptly factors that might endanger the public or the environment”. [3]  Furthermore, they must be sure as to not engage in false advertising as to the abilities of the our design or any future iterations. As this project deals with personal data the need to be clear on how accurate the readings will be and follow the third tenant, to endeavor “to be honest and realistic in stating claims or estimates based on available data”. [3]

**References**

[1] “Facts and Figures.” *Alzheimer's Association*, Alzheimer's Association, 2018,

[Accessed: 18-Sep-2018]. www.alz.org/alzheimers-dementia/facts-figures.

[2] Infrared Thermometer, MLX90614 Family Datasheet, Melexis, [Accessed: 30-Sep-2018]

https://cdn-shop.adafruit.com/datasheets/MLX90614.pdf

[3] Ieee.org, "IEEE IEEE Code of Ethics", 2016. [Online]. [Accessed: 18- Sep- 2018].

Available: http://www.ieee.org/about/corporate/governance/p7-8.html

# Appendix A

# Requirement and Verification Table

|  |  |  |
| --- | --- | --- |
| **Table 3  System Requirements and Verifications** | |  |
| Requirement | Verification | Verification status  (Y or N) |
| The battery must be able to store enough charge to provide 7.4V ±0.2V for the microcontroller. | Use a voltmeter to ensure the battery has an output voltage within the tolerance of our sensor to prevent damage to the microcontroller. | Y |
| It must have enough charge to last at least 1 hour at its ON state and additionally last 10 hours at its idle state. | Use a multimeter to measure the current the microcontroller draws from the fully charged batteries when at its ON state and idle state and using the batteries’ capacity (in mAh), calculate the time it takes for the battery to drain out. | Y |
| The battery charger must output 4.2±0.05V | Use a voltmeter to ensure the charger  regulates an output voltage within the tolerance in order to stay within the battery’s recharging ratings. | Y |
| Microcontroller supports multiple sensor integration simultaneously | Check for data from at least 3 sensors simultaneously (<1 second delay) | Y |
| The microcontroller must be able to supply 3.3±0.2V and 5.0±0.2V with respect to its internal ground. | Use a voltmeter to ensure the output pins stay within the output voltage tolerances. | Y |
| The antenna must be powerful enough to transmit data to the network 50 feet away from the router. It needs to send data packets without any packet loss or disconnections. Areas with low connectivity will be anywhere beyond 30 feet of the router and have 1 bar of wifi on a phone. | Transmit 1 sensory reading per second from the WiFi shield and antenna and confirm the database is receiving 1 multi-sensor data packet per second. | N |
| The database must be able to accept 1 multi-column input per second and log everything on the fly with a maximum 2 second latency. (No tolerance, as this is a maximum.) | Check the time delay between when a packet is sent and received. Take many samples and make sure the maximum latency is below exactly 2 seconds | Y |
| Highcharts JS must display data as interactive charts, updating every second | Test and debug software code to make sure the charts in the app are fully functional. | Y |
| The display page must have easy navigation through the application/site with readily available data. | Test and debug software code to make sure and is presented well in a mobile/web platform | Y |
| The force sensor must be able to withstand 14.5 N (maximal pinch force, men) | The FSR 402’s force sensitivity range is 0.2N - 20N, which is sufficient even for a very strong grip. Verify by testing if the force sensor can withstand our maximum pinch force (patients will most likely not pinch very hard). | Y |
| Output 1 reading per second for digital thermometer | 1. Check the periodicity using an oscilloscope  2. Confirm with our database that 1 reading has been uploaded per second. | Y |
| Accuracy within ±1˚C at room temperature for digital thermometer | Use known temperature of room to measure accuracy. | Y |
| Determine a location to place the digital thermometer on the pet where the patients will trigger it at least once in a 30-minute period | Observe patient-pet interactions to see how pets are typically held/cuddled to better place sensors. | Y |
| Measured heart rates must be accurate to ±20 BPM for IR Emitter/Detector | Compare the SEN 00241 to normal human resting heart rate levels (60-80bpm)  2. Repeat this 15 times to ensure no anomalies occur | Y |
| Output 10 readings per second | 1. Check the periodicity using an oscilloscope  2. Confirm with our database that all 10 readings have been uploaded | Y |
| IR Thermometer outputs 1 reading per second | 1. Check the periodicity using an oscilloscope  2. Confirm with our database that 1 reading have been uploaded per second. | Y |
| IR Thermometer Accuracy within ±0.5 ˚C | Measure temperature of several hot plates of known temperatures and measure accuracy. | N |
| Placed at ideal locations around the pet for better readings | Observe patient-pet interactions to see how pets are typically held/cuddled to better place sensors. | Y |