University of Illinois at
Urbana-Champaign

ECE 445 Team #14

Child Development Sensor
Final Report

Yang An Tang (ytang46), Eu Hong Woo (ewoo5),
Xiang Wen (xwen10)

December 14, 2017

Contents

1 Introduction

1.1 Objective

Human interaction at early development stages is vital to the growth of young children. It is therefore imperative that studies be conducted on the interactions between children and their parents, who are usually the most influential caregivers during early development. Modern research in child development is vast. There are countless, dynamic variables in a child's early developmental period that set children on trajectories of psychological adjustment which can be difficult to alter. Although studies are expansive, the method of current research is limited[1]. Although development is dependent on repeated real-time interactions with caregivers over large amounts of time, research observations in the lab occur in short, predetermined intervals. Development cannot accurately be assessed in isolated, static interactions. Additionally, many modern development observations occur within a controlled laboratory setting. However, growth is intimately tied to settings such as homes and classrooms. On occasions where research is conducted in natural context, assessments are brief and researchers must be present. Due to these limitations, research methodology must be advanced through innovation.

The goal of our project for Senior Design is to develop a device that allows researchers to obtain accurate data on child development. To overcome the limitations of research occurring in controlled environments, we will be developing a wireless device that can capture and transmit accurate data to a smartphone or a memory stick. This eliminates the limitation for researchers to be present during data collection. Caregivers are able to place the device on the child and conduct studies at home or classrooms, locations where multiple repeated interactions occur, as opposed to in a controlled laboratory setting. Therefore our device would enable accurate assessment of childhood behavioral and physiological development. We have been corresponding back and forth with a mechanical engineering team to design this device. The tasks were broken up into the following: We, the electrical team, is in charge of designing the sensor network, the wireless data handling/transmission, the electrical safety of the system, and the storage of the data. The mechanical team is tasked with designing the housing of the electronics, the form

factor of the sensing device, and the physical safety precautions necessary when in contact with children such as overheating and motion. This is a joint multidisciplinary effort advised and sponsored by Professor Harley Johnson, UIUC MechSE, and Professor Nancy McElwain, UIUC HDFS.

1.2 Background

The research on Childhood Development is extensive and complex due to many influencing variables in a child's environment. To obtain valuable data to assess a child's development, physiological, psychological, and behavioral patterns must be studied. Information such as a child's heart-rate variability[2][3], body temperature and movement, quality of sleep[4], and characteristics in pitch of a caregiver's voice[5][6] are all useful information in determining a child's growth and influencing factors such as stress. The main parameters we are obtaining in our Senior Design project are cardiac vagal tone, through an **ECG (electrocardiogram) waveform**, and **audio recording** through a microphone.

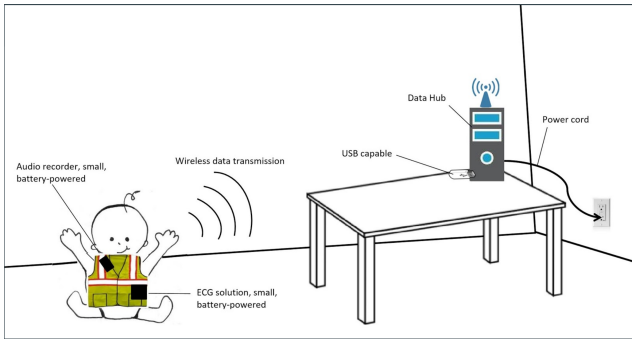

Figure 1: Use Case Scenario of Child Development Sensor System

Figure 1 gives a glimpse of what the prototype would look like. The sensor node is embedded into a vest so that an infant can simply wear the vest on top of their clothing, and all the sensors will be in contact with him/her while also making it difficult for the infant to tamper with the sensors. The sensor node in the vest sends ECG and audio recording data to the main hub, which is stationary (shown as data hub in Figure 1). The vest was designed by the mechanical team. The vest contains a fully battery powered source to allow free movement. Since everything is embedded within a vest, it would be more convenient if even the leads of the ECG will be reusable instead of the typical one-time-use disposable electric leads. Hence, when characterizing our ECG sensors, the mechanical team has looked into non-adhesive leads options, specifically capacitive electrodes. There were no major changes to our high-level or block-level system design since the beginning.

1.3 High-Level Requirements

1. Obtain accurate and synchronous ECG and audio data on research subjects.
2. Transmit sensor data through wireless in accessible formats to a cloud server so that users can access it from anywhere with an internet connection.
3. Develop and implement sensor node (ECG sensor, audio recorder, Bluetooth module, power source) and main hub (microcontroller, Bluetooth + WiFi module, power electronics).

2 Design

2.1 Block Diagram

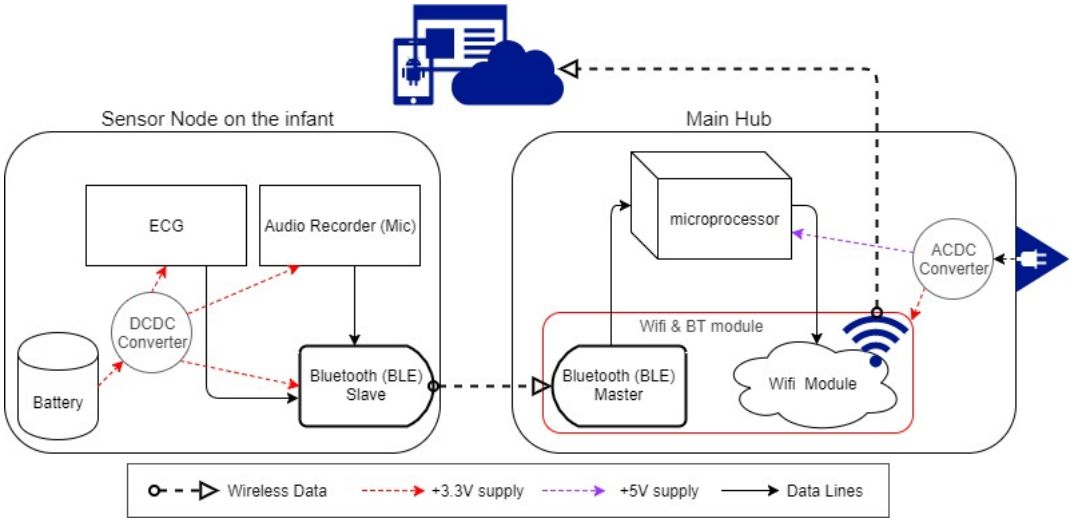

Figure 2: Full System Block Diagram

To allow an infant to move freely within a room or building while also being able to monitor their cardio signals without much obstruction, our design has to be separated into two main parts. One part will have to be worn on the infant to monitor his/her ECG and record audio. This part needs to be compact, cool (operates with low temperature) and lightweight. Thus, it needs to have minimal hardware, and the hardware needs to consume low power. So this part, which is the Sensor Node as shown in the diagram 2, only contain the necessary sensors, the audio recorder, and a low energy Bluetooth module, all powered by battery. No data processing will be done on the sensor node; instead, data streams are sent over Bluetooth directly to the main hub. The main hub is where the main processing unit lives. It contains a Bluetooth module to act as a master, a microprocessor to analyze, filter and sync both the ECG and audio recordings, and a WiFi module to push these data to a cloud server. Since the main hub can consume more power, it draws power from the wall. Detailed specifications of each block will be discussed further in the following sections.

2.2 Sensor Node Functional Overview

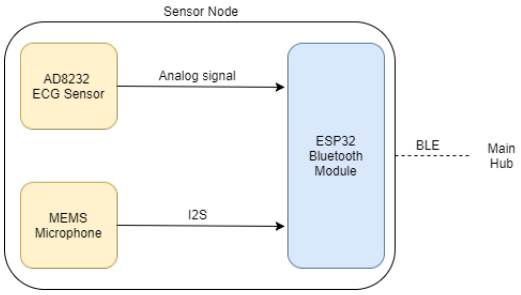

Figure 3: Sensor Node Modular Block Diagram

The Sensor Node shown in Figure 3 is the components that will be attached to the child during experiments. This is the main sensing component that communicates over wireless with the main

hub through Bluetooth. Subcomponents contained within the sensor node include an ECG sensor, MEMS microphone, power electronics, and communication antennas. In Figure 3, Labeled are the chosen components/modules for our implementation.

Firstly, the ECG sensor records a child's heartbeat as a waveform instead of beats per minutes (bpm). To accomplish the above, we have to decide how many electrodes are necessary or sufficient to graph usable waveforms. Most common ECG lab machines have 12 leads. Using 12 leads would provide high accuracy readings and can be used to diagnose heart diseases, but this would make our final package too bulky for a toddler to wear. Alternatively, we could use 3 or 5 leads. Lesser electrode contacts result in noisier waveforms and less accurate measurements. That being said, given that the use of these ECG waveforms is just for monitoring a child's heartbeat instead of diagnosing heart diseases, we have used just 3 leads.

The audio recording aspect is to supplement ECG data such that researchers would know what the infant/toddler is doing during the measurement period. The biggest challenge would be implementing a microphone module that is sensitive for clear audio recording, eg. speech. Both these sensors communicate directly through data lines or buses with the Bluetooth module. Given the selected sensor modules, AD8232 ECG module outputs analog signals which are then sent to the ESP32 module's built in ADC before being packaged and transmitted. For the MEMS microphone, it outputs digitally through I2S ports. The Bluetooth module, ESP32 happens to be the right choice since it has all the necessary I/O ports.

The full circuit schematic for the sensor node can be found in the Appendix figure 29. Each submodule within the schematic will be touched on in later parts of the report in detail.

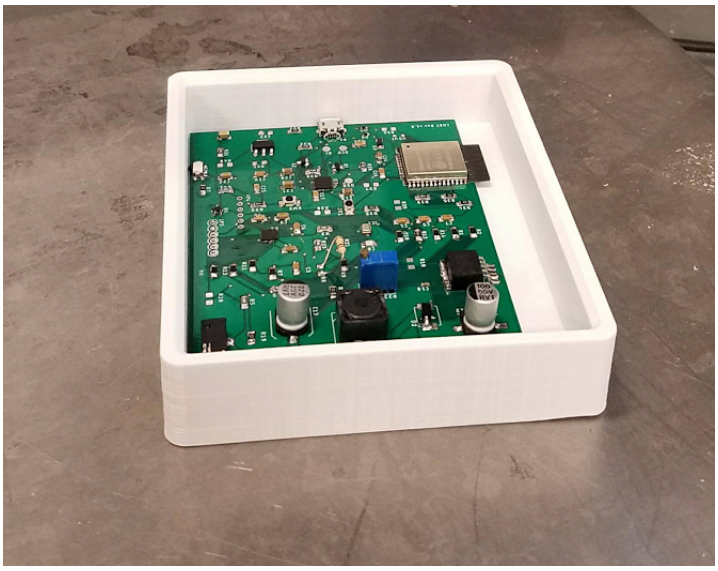

Figure 4: Current sensor node prototype in housing

Figure 4 shows the final PCB of the sensor node within the housing provided by the mechanical team. The overall weight is about 0.2 lbs which satisfies the form factor requirements.

Sensor Node - ECG Sensor

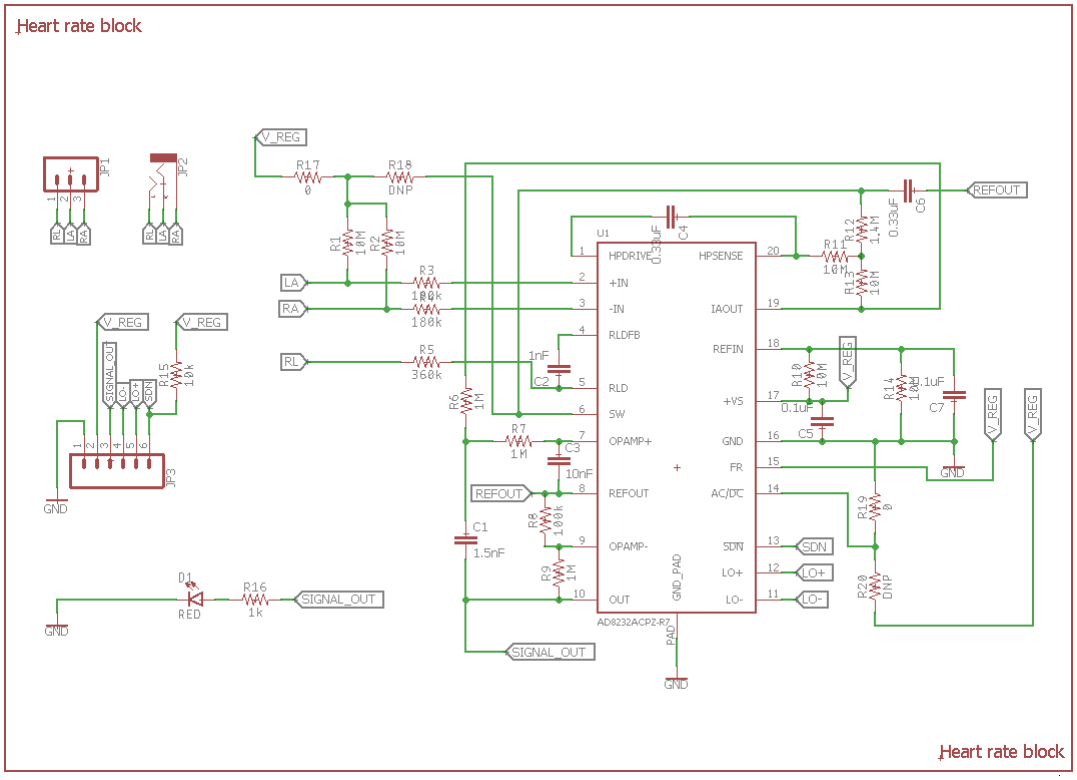

Figure 5: ECG sensor block schematic

Typical voltage differences throughout our body caused by our heartbeats are in the order of a few hundred microvolts to a few milivolts [17]. We use three leads since the accuracy we are aiming to achieve is only about 70% as requested by the research team. This posts some challenges since having lesser leads might decrease the signal to noise ratio. For higher resolution, we can obtain 16-bit sample at a sampling rate of 125 Hz to 500 Hz [18]. The ECG sensor that we are using is the AD8232 from Analog Devices which outputs an analog signal. Figure 5 shows the complete circuit design encompassing the AD8232 chip, note wire `SIGNAL_OUT` which carries the analog signal that goes to the ESP32 Bluetooth module. Looking at the schematic, there are input pins `OPAMP+` and `OPAMP-` used to control output signal gain.

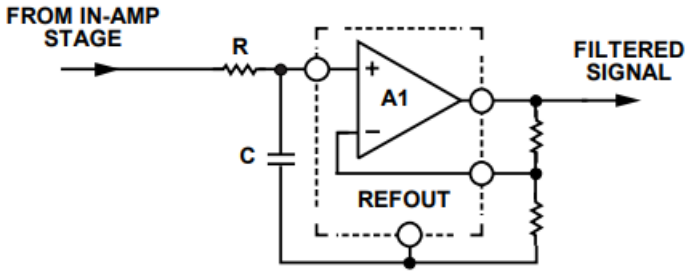

Figure 6: Schematic for a Single-Pole Low-Pass Filter and Additional Gain

The AD8232 includes an uncommitted op amp that can be used for extra gain and filtering [11]. Since our application does not require a high-order filter, a simple RC low-pass filter should suffice, and the op amp can buffer or further amplify the signal. Based on figure 6, there are two resistors to the right of the opamp (operational amplifier). Modifying these resistors control the

output gain.

$$Gain = 1 + \frac{R1}{R2} \tag{1}$$

where R1 is the top resistor and R2 is the bottom resistor. In our configuration, we have chosen $R1 = 1\ M\Omega$ and $R2 = 100\ k\Omega$ which gives us

$$Gain = 1 + \frac{1\ M\Omega}{100\ k\Omega} = 11 \tag{2}$$

with a gain of 11, the output signal amplitude is large enough such that R-peaks are clearly distinguishable.

Figure 7: ECG sensor acquired waveform

Using the on-board ESP32 module's Analog-to-Digital (ADC) converter which has a **sampling depth of 12-bits** and with a **sampling frequency of 55 Hz**, the output waveform in figure 7 shows very distinguishable R-peaks which can correspond to beats-per-minutes (bpm) using the following equation,

$$bpm = \frac{60\ s}{RR\ peak\ interval\ (s)} \tag{3}$$

where RR peak interval is the average time interval between R-peaks.

ECG Data - Research			ECG Data- CDST (fs = 200 Hz)			
Time(s)	RR-I (s)		Sample	Amplitude	RR-I (samples)	RR-I (s)
0.502	0.709		80	2020	144	0.72
1.211	0.702		224	1994	142	0.71
1.913	0.697		366	2016	140	0.7
2.61	0.691		506	2003	140	0.7
3.301	0.691		646	2011	138	0.69
3.992	0.696		784	2028	138	0.69
4.688	0.659		922	1988	139	0.695
5.347	0.727		1061	1965	138	0.69
6.074	0.668		1199	2065	140	0.7
6.742	0.713		1339	1910	139	0.695
7.455	0.684		1478	1998	136	0.68
8.139	0.736		1614	2035	143	0.715
8.875	0.772		1757	2033	148	0.74
9.647	0.789		1905	2127	154	0.77
10.436	0.756		2059	2077	157	0.785
11.192	0.757		2216	2035	152	0.76
11.949	0.766		2368	2038	151	0.755
12.715	0.716		2519	2119	153	0.765
13.431	0.73		2672	2081	143	0.715
14.161	0.673		2815	1961	141	0.705
Average =	**0.7166**		2956	2081	Average =	**0.719**

Figure 8: ECG sensor comparison with current research setup

In the above figure 8, table on the left shows average RR peak interval for the current research setup and on the right is our system's average interval. These data points are recorded from 15 seconds of ECG data using both the researcher's setup and ours on a test subject simultaneously

using only three electrode leads each. Using equation (3),

$$Research\ setup = \frac{60\ s}{0.7166\ s} = 83.729\ bpm \tag{4}$$

$$Our\ system = \frac{60\ s}{0.719\ s} = 83.449\ bpm \tag{5}$$

our new sensor system shows a $< 1\%$ error when taking the research setup as a baseline which is much lesser than the required minimum of 30% error.

We have shown that our ECG sensor satisfies all of the corresponding requirements and verifications as shown in table 6 row one. R-peaks are easily distinguishable, and ECG sensor is more than 70% accurate when compared to the current research equipment.

Sensor Node - Audio Recorder

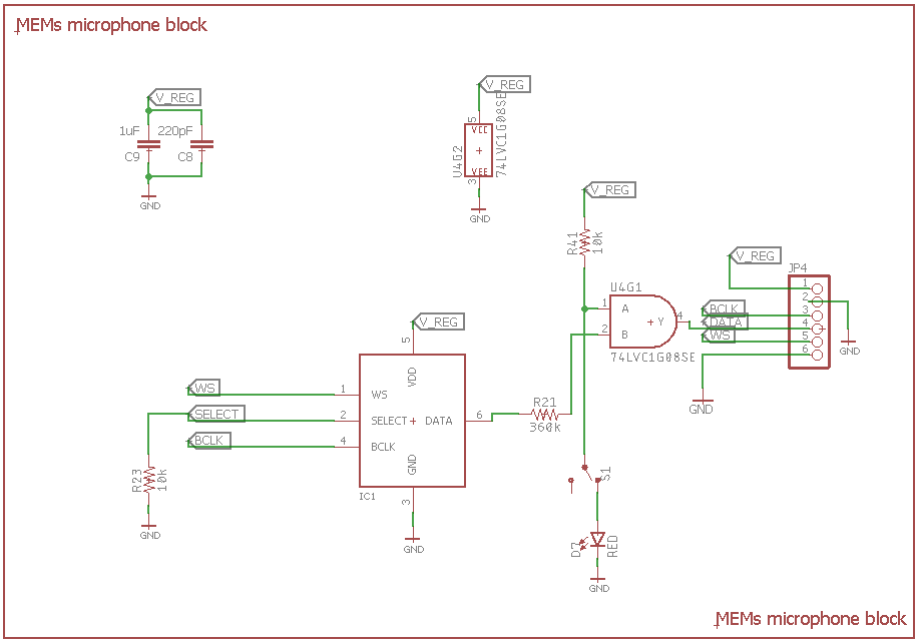

Figure 9: Audio recorder block schematic

We were initially considering two forms of microphones, electret and MEMs. Electret microphones have a typical frequency range of 20 Hz - 20 kHz which is more than sufficient for our needs as we are trying to record voices at most, but an electret microphone might not be sensitive enough such that audio recordings will be clear. On the other hand, MEMs microphones are smaller and more compact in size. Also, they offer better audio reproduction that are less prone to vibrations compared to electret microphones. Another advantage to MEMs microphones is the fact that they produce digital output instead of analog outputs which make interfacing with microcontrollers much easier. Thus, we have decided to use the SPH0645LM4H MEMs microphone which produces digital output in accordance with I2S protocols. It has a sampling rate of 32 kHz - 64 kHz, a bit depth of 18-bits, a SNR of 65 dB-A, and a sensitivity which allows audio pitches around 50 Hz - 15 kHz before clipping [19].

The full schematic for the MEMs microphone implementation is shown in figure 9, note wire `DATA` which carries the digital I2S signal that goes to the ESP32 Bluetooth module. Each component on the schematic that requires power is connected to the common rail `V_REG` that carries 3.3

V. Within this circuit, there is a mute switch for privacy purposes in the form of an AND gate where one of the input comes from the microphone and the other input comes either from `V_REG` or ground depending on the switch state. A LED diode was added to the circuit that lights up when microphone is muted.

As mentioned earlier, the microphone provides 18-bit depth of audio, however, since we are limited by the Bluetooth and UART transmission speed, we have decided to configure the microphone to operate with 3 kHz sampling rate and use the 16 MSBs (Most Significant Bits) of the samples. The way we have decided on these values is by calculating our limit based on the maximum baud rate of the UART port and Bluetooth transmission speed. Our maximum baud rate is 115200 bits/s, so we can approximate our data transfer rate by calculating the bit rate with this sampling rate and bit depth:

$$Bit\ rate = 3000\ samples/s \times 24\ bits/sample = 72000\ bits/s \tag{6}$$

This is way below the 115200 baud rate limit, which implies we can actually go for higher quality audio. Unfortunately, through an iterative process, we found out that even though the ESP32 promises a BLE bit rate of 4 Mbps[7], the best we can achieve is sending a 20 character array each time, with a rate of about 500 Hz. Even with optimized algorithms, we were only able to send audio data through Bluetooth at 3 kHz without dropping samples.

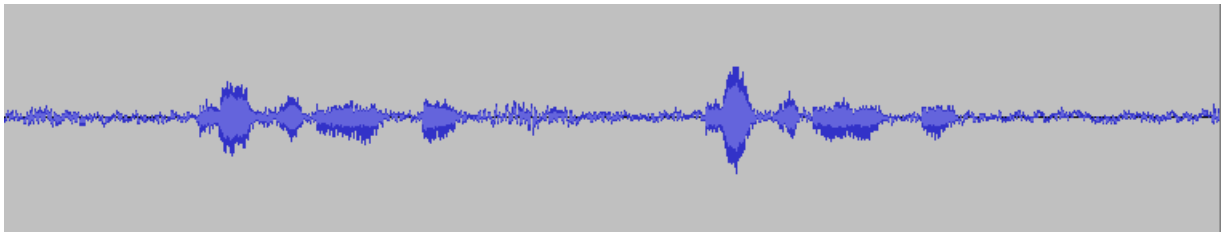

Figure 10: Output waveform of our sample audio

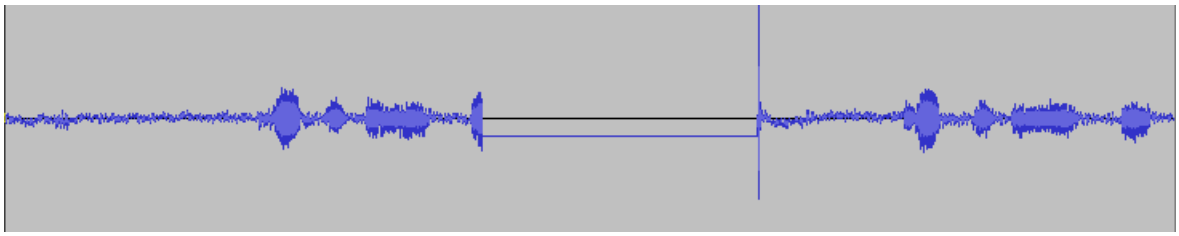

Figure 11: Output waveform of our sample audio with mute switched during the interval

Figures 10 and 11 above are audio samples obtained using the microphone on our own PCB and received through Bluetooth. In figure 11, we can see that when the mute button is switched, the audio is flattened out, effectively satisfying the hardware mute requirement laid out in Table 6. For the audio samples in both figures, the audio looks noisy. We have tackled this problem just by manually selecting our bit depths and offsets. Due to the sensitivity of the microphone on our PCB, voices get amplified above the clipping threshold. Our remedy to this is to lower the amplitudes by right shifting the audio samples. We know this is not the right method as each audio signal amplitudes are shrunk a different amount. Unfortunately, because of time constraints, we did not venture into digital filtering methods, resulting in muffled audio which does not satisfy the SNR requirements in table 6. That being said, since our main objective is only to get audible voice recordings, our microphone definitely does the job.

2.3 Main Hub Functional Overview

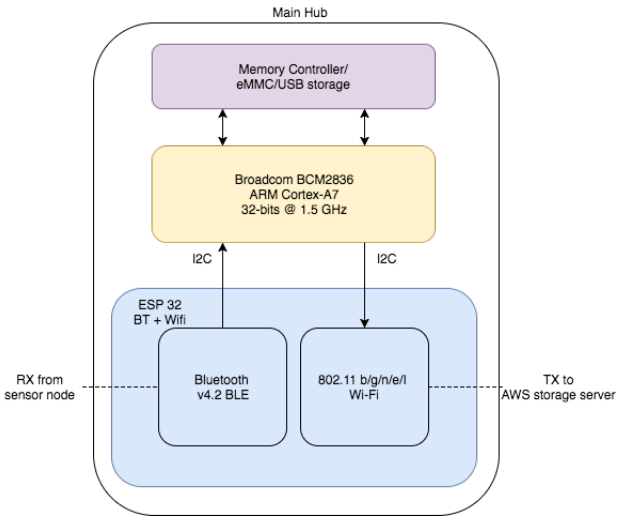

Figure 12: Main Hub Modular Block Diagram

The main hub is the processing unit of our system. All of the sensor node data will be transmitted across Bluetooth and then processed by the microprocessor before being pushed either to both local memory and cloud storage via WiFi. We chose to have most of the high consumption components away from the child to prevent any unwanted harm. Figure 12 shows the modular breakdown of the components in the main hub which consist of a microprocessor, BT + WiFi module, and some form of memory storage. Here we have chosen for each component what we think suits our needs best. A Broadcom BCM2836 ARM A7 microprocessor, ESP32 Bluetooth and WiFi combo module. In terms of storage, ideally all collected data points will be pushed to some form of cloud storage, in this case AWS DynamoDB. But for debugging and prototyping purposes, data is also stored in local storage.

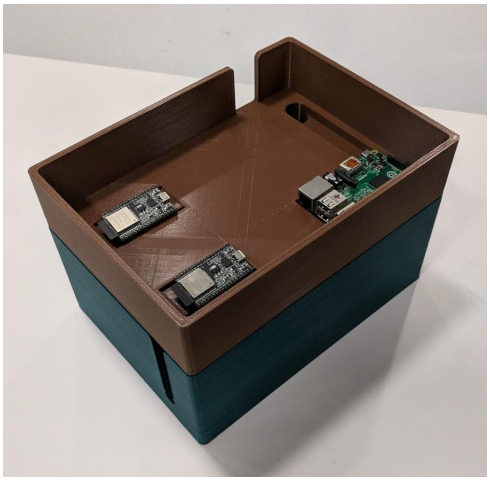

Figure 13: Current main hub prototype in housing

Main Hub - Firmware

There are a few things on this project which required us to write some of our own firmware. For the I2S protocol needed on sensor node's ESP32 chip to receive audio samples, we needed to configure the port such that incoming samples are treated as unsigned 32 bit samples. This is the case even though we mentioned that our microphone chip only provides 18-bit depth because

of how the chip's DMA (Direct Memory Access) buffers deal with incoming data. For UART protocols needed on the main hub to communicate between the ESP32 chips and BCM2836, we had to use default C Linux system header files to open and configure UART serial ports.

After all the communication protocols are configured properly, we needed to encode and decode our data samples for two reasons:

1. Bluetooth and WiFi transmission speed is limited, and data transfered are treated as characters or strings. So, instead of printing the decimals for each sample (each digit uses up one byte, very inefficient), we needed to compress and encode into character strings that can be decoded later.

2. Character or strings are encoded using the UTF-8 standard, so there will be characters which we will need to avoid sending, i.e. DEL, ESC, new line, NULL and so on.

With all these things in mind, we came up with our own data encoding for both ECG and audio samples:

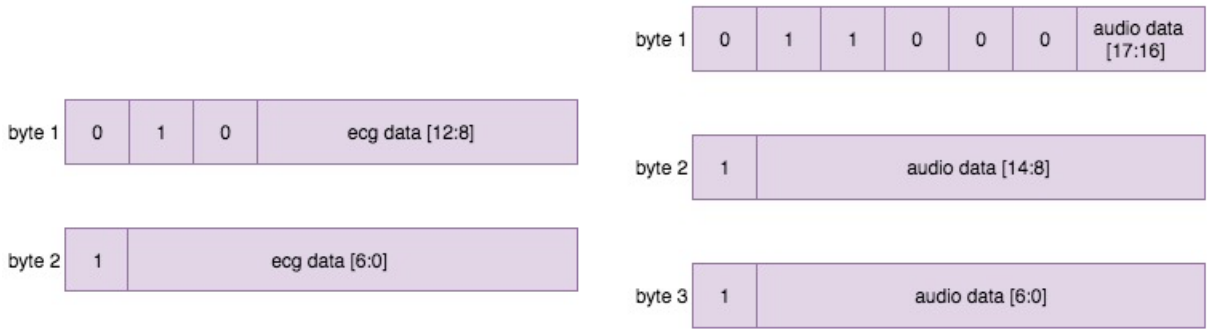

Figure 14: Data encoding for ECG samples (left) and encoding for audio samples (right).

Using the data encoding shown in Figure 14 above, we can distinguish the first byte in the sequence. After receiving the first byte, a bit mask on the 3rd MSB can be used to deduce if the incoming data is from ECG or microphone. On the transmitter's side, the 3rd MSB needs to be set to '0' if it is an ECG sample, and '1' if it is an audio sample. The 2nd MSB is set as '1' in both cases because we need to avoid the first 64 UTF-8 characters. Bit indexes 2 to 4 must be set to '0' for audio samples because we need to avoid DEL (delete), which is hex 0x7F. This way of encoding limits each audio sample to 3 bytes (or 3 characters) and each ECG sample to 2 bytes, which allows us to fit 6 audio samples in one Bluetooth transmission (maximum of 20 characters), and still be able to occasionally fit a single ECG sample. Using this allowed us to utilize the Bluetooth to send audio at 3kHz without dropping samples.

Another firmware issue which we had to face is the conversion of USB to serial signals. In order to still be able to flash the ESP32 chip on our PCB, we used a CP210x chip for this conversion. Because of some firmware bug, we had to manually short DCD and RI (connections for serial RS-232) to signal a new connection to the host computer, then detach DCD when the connection has been established.

2.4 Wireless Transmission

Wireless Transmission - Bluetooth

Initially, we chose to use different Bluetooth modules for the sensor node and main hub because we do not need a WiFi capable module on the sensor node. But we have since resorted to using

the same ESP32 board for both, so that we can save on hardware and software development time.

Firstly, to make sure that the ESP32 hardware provides **sufficient throughput/bandwidth** for our purposes, a simple calculation was done. Each `Write Command` allows 20 Bytes of data according to specifications. These packets do not provide any application level acknowledgement of sending, and hence can be queued. Next, the lowest interval allowed is 7.5 ms and the maximum number of packets per connection event is six.[7]

Given number of packets per event as n, connection interval T, and a single command allows B number of Bytes, the throughput is

$$Throughput = n \times B \times \frac{1}{T} \tag{7}$$

where $n = 6$, $T = 7.5\ ms$, and $B = 20$, total throughput is

$$Total\ Throughput = 6 \times 20 \times \frac{1}{0.0075\ s} \tag{8}$$

$$= 16\ kB/s = 128\ kbps \tag{9}$$

which is more than enough for our system's usage. Microphone data will be using up most of the bandwidth as it is running at 3 kHz in order to capture audible speech.

Range and sensitivity is also another major consideration for our purposes. Since BLE on sensor node will be transmitting and BLE on main hub will be transmitting and receiving respectively, it is important that the line-of-sight range is at least the span of an average size room in order for the connection to work.

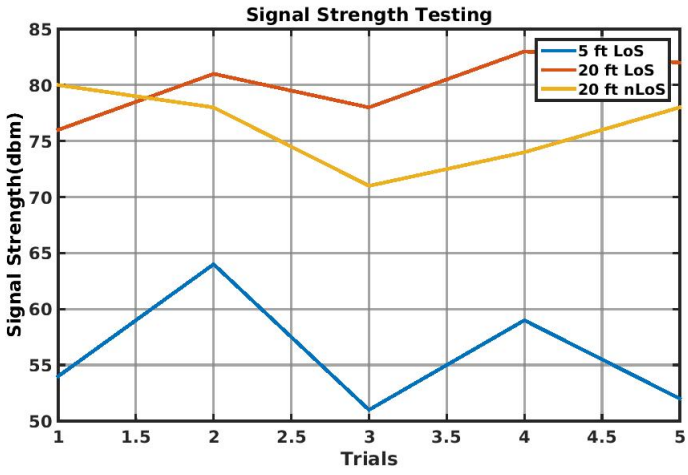

Figure 15: Sensor node Bluetooth experimental range

Based on figure 15, range testing was done at three different distances from the master which is the main hub in this case. Three cases are, 5 feet line-of-sight (LoS), 20 feet LoS, and 20 feet non-LoS. Each case was repeated 5 times to take an average. Results were very promising and do meet requirements, 5 feet LoS had a signal strength of -55 dBm, 20 feet LoS at -80 dBm, and a 20 feet non-LoS at -75 dBm.

We have shown that our Bluetooth implementation satisfies all of the corresponding requirements and verifications as shown in table 6. The system works at a distance of 20 feet LoS and non-LoS

with a sensitivity better than -70 dBm.

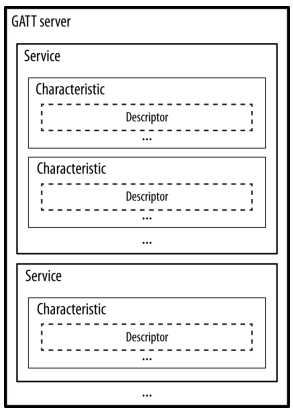

Figure 16: BLE GATT Server Structure

For the ESP32 firmware, a GATT server had to be setup according to BLE protocols for the sensor node and main hub to communicate. GATT server specifications can be found at [10]. The main hub in this case will be referred to as the master and the sensor node as the slave. In order for the slave to write to the master, a specific Service UUID acts as the "address" at which both the master and slave will be communicating at. The master will have to setup a GATT Server that hosts the specific Service UUID so that the slave can connect to it. Within a given service, there can be more than one characteristic which in this case denotes data from sensors. Since we have both ECG and microphone sensors, there are two characteristics with unique given Characteristic UUIDs under the same Service.

During development, there were a few bugs, namely scanning took a long time due to too many Bluetooth devices in the area and the written embedded application kept crashing. To fix the first problem, hardware scan rate was increased through the BIOS. For the second problem, the user-program stack size was increased and heavy standard C libraries were omitted.

```
/* BLE Server initialization, advertising server, acknowledging client connection */
BLE.init()
BLE.create(server)

/* create server with service of "serviceUUID" and characteristic of "charUUID" */
BLE.server.create(serviceUUID)
BLE.server.create(charUUID, value)

/* start advertising server */
BLE.advertising(start)
```

Figure 17: Pseudocode for BLE Master.

```
/* BLE Client initialization, connecting to server, and writing to specific service */
BLE.init()
while(1)
    if(BLE.scan(serviceUUID))
        BLE.create(client)
        /* check to make sure service exists */
        if(!BLE.connect(serviceUUID->address))
            print "Failed to connect to server"
            return
        /* writes to service->characteristic->value */
        BLE.write(address->charUUID, value)
    else
        print "Failed to find server"
```

Figure 18: Pseudocode for BLE Slave.

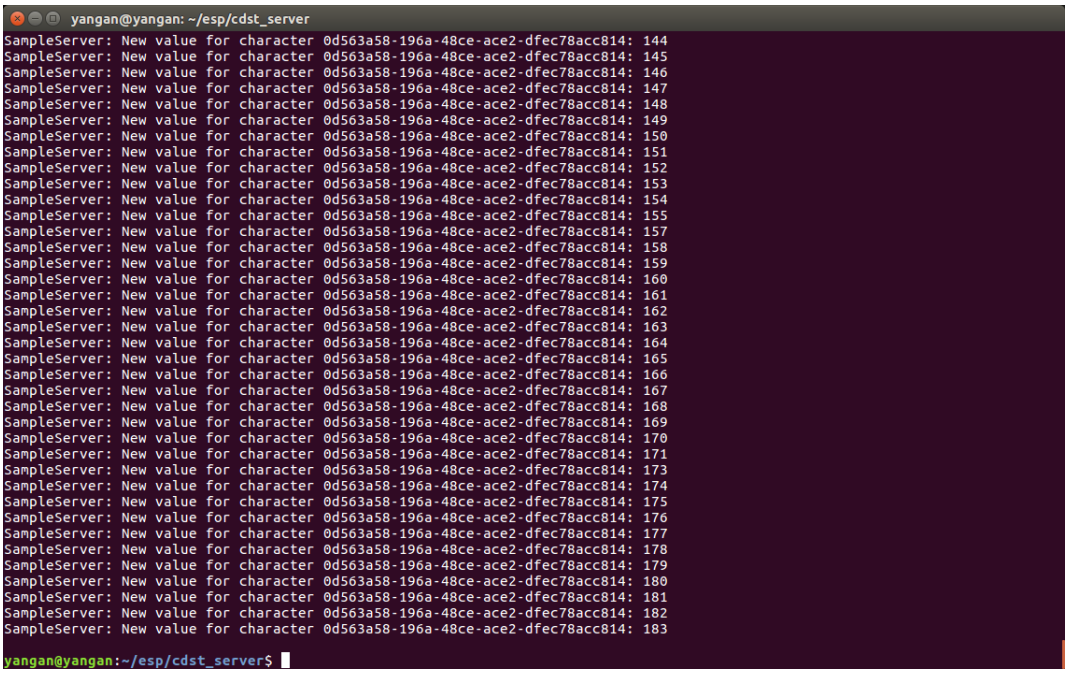

Figure 19: BLE Master successfully hosting server and receiving each packet from Slave.

In figure 17 is the pseudocode for initializing and advertising a BLE GATT server. Once initialized, the program waits for an incoming connection request from the specified service UUID. On the other hand in figure 18, is the pseudocode for initializing a slave then scanning for the specified service UUID before requesting to connect. Figure 19 is a screenshot of the server's output showing that values are received successfully.

Wireless Transmission - WiFi

WiFi is only enabled on the ESP32 chip that is on the main hub and connected to the microprocessor. The general idea is for the ESP32 chip to read data over serial from the microprocessor, then send the TCP packets using MQTT protocols to an AWS IoT server. To establish MQTT connection, the WiFi client must first request connection with a unique client ID, once connected the client can start publishing to the specific topic that is setup by the server. A flowchart of the data pipeline is shown in the Appendix 6.

```
/* Pseudocode: WiFi connect to AP and send to AWS IoT using MQTT protocol */

/* Initialize WiFi event handler and connect to AP */
wifi.init()
if(!wifi.connect(AP))
    print "Failed to connect to access point: %x", AP
    return

/* Connect and initialize MQTT channel */
if(!AWS.connect(clientID))
    print "Failed to connect to AWS server of ID: %x", clientID
    return
AWS.init()

/* Read from microcontroller using I2C protocol
 * Publish or write to specific MQTT topic
 */
while(1)
    value = I2C.read(microcontroller)
    AWS.publish(value)

AWS.disconnect()
```

Figure 20: Pseudocode for WiFi algorithm that publishes to AWS Server.

The above pseudo-algorithm in figure 20 is very simple compared to the actual implementation because it omits most of the configuration declaration such as callback functions, hardware pin settings, drivers, Rx/Tx buffer, TLS structure etc. Callback functions in this case will be invoked once a Tx packet is sent, e.g. to read next value in serial buffer. Next, the Rx/Tx buffer acts as a queue since the program has to wait for a receive acknowledgement before sending the next packet. A buffer queue is needed to maintain First-In-First-Out (FIFO) arrangement.

Collected data will be pushed to the AWS cloud server using AWS IoT API calls. This is automated by using AWS Lambda which runs a Python script that adds timestamps then pushes to AWS DynamoDB. AWS IoT helps with the MQTT handshake which is a lightweight messaging protocol optimized for high latency. To further explain, MQTT is a machine-to-machine (M2M)/"Internet of Things" connectivity protocol. It was designed as an extremely lightweight publish/subscribe messaging transport [12].

We faced a problem when enabling both WiFi and Bluetooth on the ESP32 hardware. The embedded firmware crashes as if there is not enough stack space. The kernel code runs fine when either WiFi or Bluetooth is enabled one at a time. Increasing stack space works only occasionally because every new call to the WiFi or Bluetooth class adds to stack space. A couple of probable solutions were tried such as going through the libraries and omitting those that are not used, creating a virtual memory map so that every new task will not override kernel code. Unfortunately, we were not able to sort this out, instead two separate ESP32 chips were used in the main hub, one for Bluetooth and one for WiFi. Ideally we would want to consolidate both Bluetooth and WiFi into a single chip, but having it separate did not affect the system's functionality. Data can still be pushed to the cloud without any hiccups satisfying the Wireless R&V as shown in table 6.

2.5 Power Electronics

Power Electronics - Sensor Node

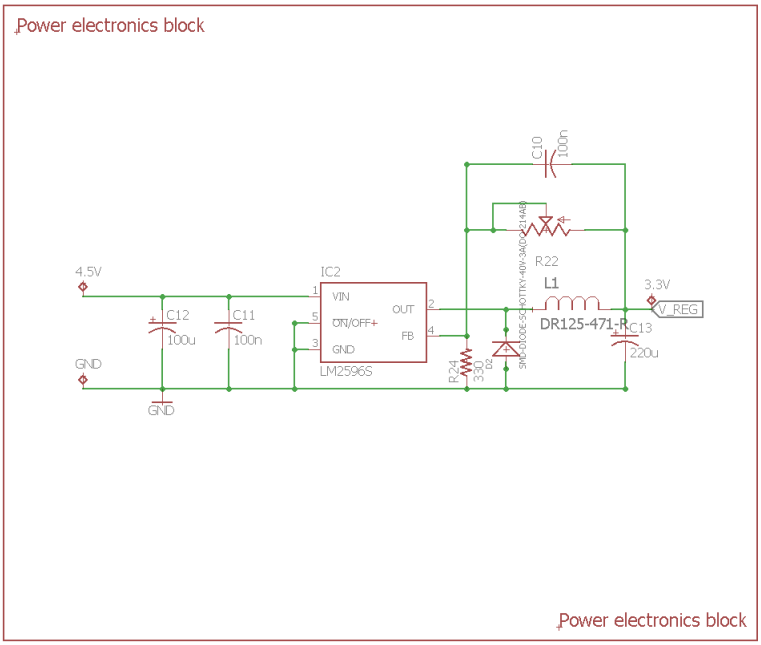

Figure 21: Buck converter schematic

The sensor node is powered by a DC-DC, buck topology, converter designed around the LM2596 chip. The chip serves as the switching and control modules of a buck converter. Offering a wide input range of 3.7 V to 40 V and capable of supplying loads of up to 3 A, the LM2596 is very versatile and used in not only the sensor node but also in the main hub which will be discussed later. Using feedback circuitry, the LM2596 can also provide an adjustable output voltage, which allows quick design changes. Alternative design options would include using other switching modules or designing the buck converter from scratch. These options were simulated and explored theoretically early in the design process; however, these were not pursued due to the many benefits of the LM2596.

The main requirement of the buck converter is to supply, on average, 3.3 V to the Bluetooth module[9], microphone [8], and ECG chip[11]. The operating ranges of all components of the sensor node lie between 3.0 V and 3.6 V as stated by their respective datasheets. Switching converters cannot provide a constant DC voltage due to the finite values of the filter inductor and capacitor. Therefore, the converter must be designed with a voltage ripple specification of +/- 0.3 V from the 3.3 V average. Knowing this ripple specification, the inductor and output capacitor can be selected correspondingly large to maintain the output voltage of the buck converter within these limits. In a typical buck converter, the minimum values of L and C are given as:

$$L = \frac{D(1-D)Vin}{f_{sw}\Delta i_l}, \; C = \frac{\Delta i_l}{8 f_{sw} \Delta v_c} \tag{10}$$

where f_{sw} corresponds to the switching frequency and Δi_l and Δv_c corresponds to the ripple of inductor current and capacitor voltage.

These equations were used in the initial design phase to select appropriate values for the components. The final values of these components were chosen according to the suggestions from the application notes of the switching module. Likewise, the input capacitor and the feedback circuitry were designed as guided by the documentation[21]. The schematic for the buck converter can be seen in figure 21.

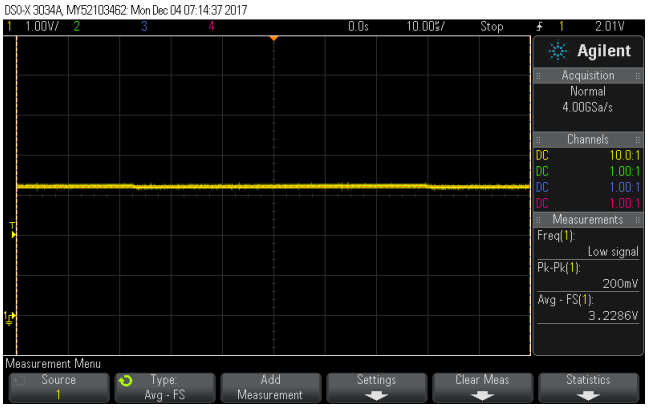

Figure 22: Output Voltage Waveform

The output voltage of the buck converter was captured on an oscilloscope and the ripple is measured using the peak-to-peak measurement function on the scope. The peak-to-peak reading must be less than 0.6 V. This is shown in figure 22. It was confirmed that both the 3.3 V average and the +/- 0.3 V ripple specifications are met. Minuscule variations in voltage reading can be attributed to losses in the measurement equipment and can be disregarded.

The converter must also supply power to the components for at least 3 to 4 hours, a typical duration of one research session. Swapping out batteries mid-session is undesirable. The operation time of the sensor node can be determined by computing the sum of typical current draws of each component, also known as the output current, I_{out}. This is calculated to be:

$$\begin{aligned} I_{out} &= I_{BT} + I_{mic} + I_{ECG} \\ &= 80\ mA + 170\ uA + 600\ uA = 81\ mA \end{aligned} \tag{11}$$

The average output power can then be calculated to be approximately $P_{out} = V_{out}I_{out} = (3.3\ V)(81\ mA) = 0.267\ W$. The input current can be calculated using efficiency, $\eta = \frac{P_{out}}{P_{in}}$. The efficiency of the converter was determined to be approximately 40% by connecting the converter to a DC supply input of 4.5 V and an equivalent load resistor, $R_{load} = \frac{3.3\ V}{81\ mA} = 40\ \Omega$. Using a 4.5 V battery source, constructed by three AAA alkaline batteries in series, the input current, I_{in}, is calculated by:

$$I_{in} = \frac{P_{out}}{\eta V_{in}} = \frac{0.267\ W}{(0.4)(4.5\ V)} = 150\ mA \tag{12}$$

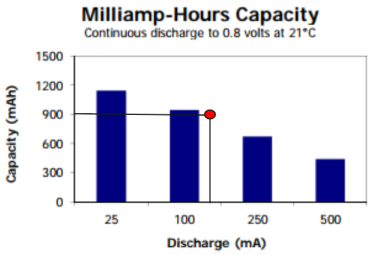

Figure 23: Point of Operation

With the input current known, the datasheet of the Energizer E92 batteries[22] can be referenced to determine the corresponding capacity. Figure 23 shows the capacity at a constant 150 mA discharge rate to be approximately 900 mAh.
The theoretical operation time, $OT_{theoretical}$, can then be calculated to be:

$$OT_{theoretical} = \frac{Capacity}{Discharge} = \frac{900\ mAh}{150\ mA} = 6\ hours \tag{13}$$

The actual operation time was verified through the process of attaching an equivalent load across the battery. Knowing the current draw of 150 mA, the equivalent resistance can be calculated to be $R_{eq} = \frac{4.5\ V}{150\ mA} = 30\ \Omega$. The buck converter is known is shut off completely at 3.7 V and will not provide power to the rest of the components. Therefore the time the batteries take to deplete to this voltage level can be assumed to be the operation time.

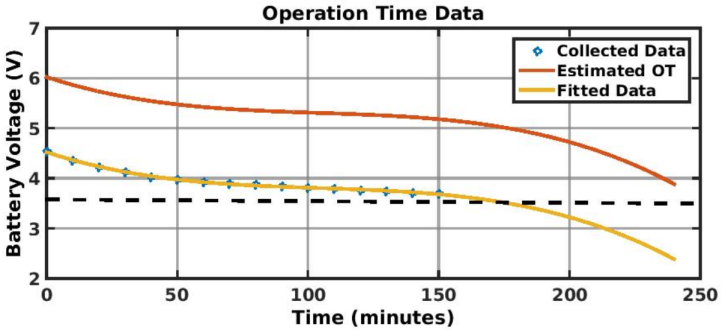

Figure 24: Operation Time Testing and Analysis

Figure 24 shows the collected data points of a 4.5 V battery setup along with a polynomial function fit of the data set. It can be seen that with three batteries, the operation time of the circuit will be approximately 2.5 hours. To overcome this lifetime issue, simply adding a fourth battery in series will bump the operation time to a little over 4 hours which will satisfy the operation time requirement. Due to the wide input range of the LM2596, the buck converter will continue to operate with a 6 V source seamlessly. The figure 24 contains an expected operation time for four batteries and satisfies the operation time requirement shown in table 6.

Power Electronics - Main Hub

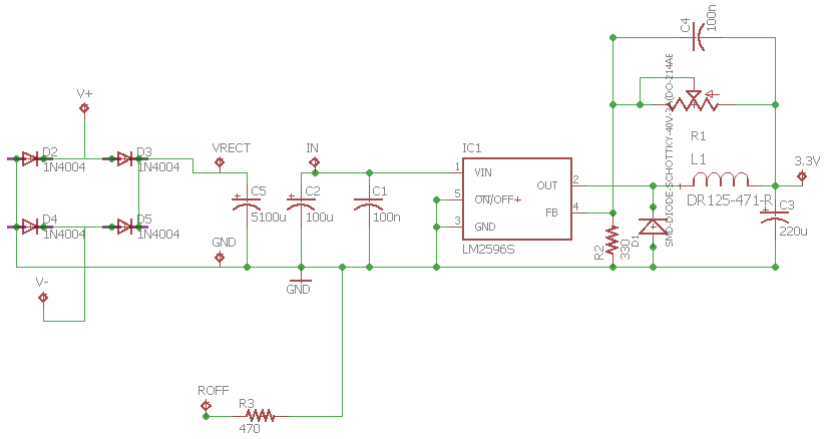

Figure 25: AC-DC Converter

The main data hub was designed to be a stationary component that will be plugged into a wall outlet. Assuming the product will be used in the United States, the outlet provides 120 Vrms at 60 Hz. Therefore AC voltage must be converted to DC for circuit applications in the main hub. The hub contains various components that need DC voltage which include the communication modules for WiFi, Bluetooth and the processor.

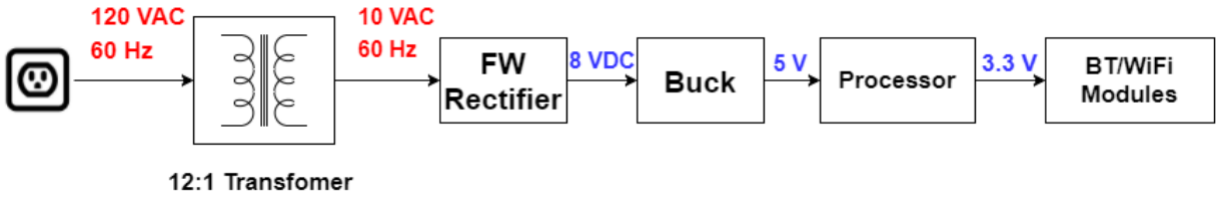

Figure 26: AC-DC Conversion Process

Once the AC voltage is rectified into a DC voltage, DC-DC conversion is used to step down the voltage to the level the main hub components will be able to use. The block diagram of the conversion process is seen in figure 26. As seen in the block diagram, the wall outlet is first connected to a transformer which steps the voltage down from 120 VAC to 10 VAC, characteristic of a 12:1 transformer. The now-lowered AC voltage becomes the input to a full-wave rectifier. A full-wave topology is used, instead of a half-wave, to provide higher average voltage and to reduce the sizes of the filtering components. The output capacitor filters out the AC voltage and provides a DC signal of approximately 8 V due to two diode drops in the rectifier. A buck converter, based on the same LM2596 chip used in the sensor node, is used to provide 5 V to the microprocessor. The processor contains built-in linear regulators which provide 3.3 V. This is used to power the WiFi and Bluetooth modules.

Design on the power electronics of the main hub began by understanding the power requirements. In our main hub, there are three components: a microprocessor, Broadcom BCM2836, and two Bluetooth and Wifi modules, ESP32. The maximum power rating for the microprocessor can be calculated as $(400\ mA)(5\ V) = 2.0\ W$. The maximum power load for the communication modules can be calculated to be $(80\,mA)(3.3\ V) = 0.264\,W$. See the total load power calculation, P_{out} in equation (14).

$$\begin{aligned} P_{out} &= 2P_{ESP32} + P_{micro} \\ &= .528\ W + 2.0\ W \\ &= 2.528\ W \end{aligned} \tag{14}$$

Because the main hub is plugged into an outlet, this maximum power load is not be a problem.

Figure 25 shows the schematic of the power electronics on the main hub. The filter capacitance is given by the equation:

$$C = \frac{I_{load}}{2f\Delta v_c} \tag{15}$$

where Δv_c is the voltage ripple, constrained to be 0.6 V, and f is the frequency of the AC source, 60 Hz.

Notice that there are various test points in the schematic which are used to connect a mechanical switch. This is to deplete the charge across the capacitor when the circuit is unplugged, an on-off switch. The depletion resistor was chosen to provide a quick discharge time with appropriate power ratings. Using the adjustable output voltage mode and wide input range of the LM2596, a 5 V output was produced from a 8 V input.

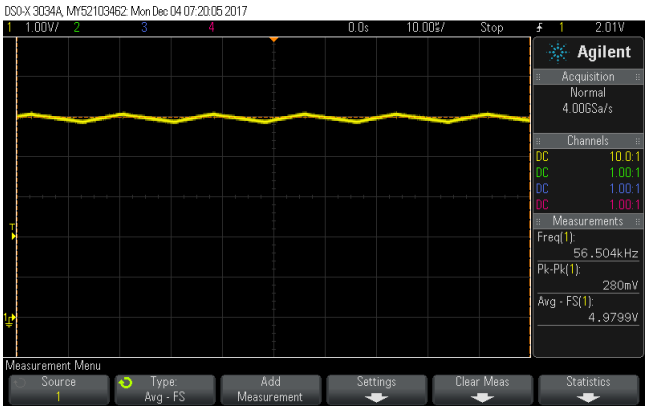

Figure 27: Output Voltage (5 V) of Converter

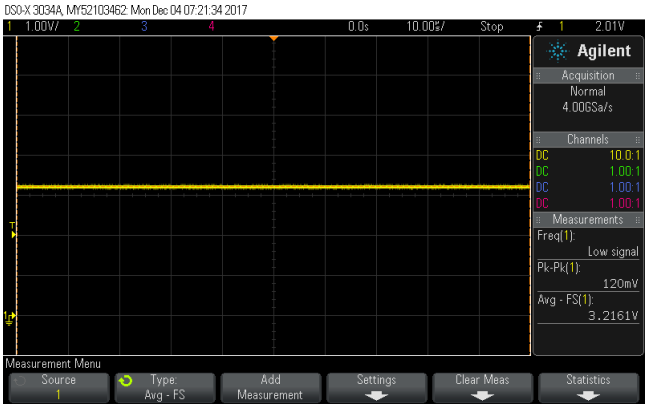

Figure 28: Input Voltage (3.3 V) to Communication Modules

Figure 27 shows the output voltage of the converter meeting specifications of 5 V with +/- 0.25 V ripple. The ripple specification is documented in the datasheet of the processor. Figure 28 shows the output of the voltage regulators on the processor which provides the 3.3 V satisfying the power requirements for both communication modules.

3 Cost

Item	**Cost**
Labor	
$32.21 (per hour)[23] x 8 (hours per day) x 30 days (per month) x 3 (labor)	$23,191.20
Parts	
MEMs Microphone (SPH0645LM4H) ECG sensor (BMD101) Electrode Cable (bundle of 3) Bluetooth & Wifi Module (ESP32) Microcontroller (MYS-437X-100-C-S) 2-sided PCB (sensor node and main hub)	$6.95 $69.00 $4.95 $30.00 = 2 x $15.00 $129.00 $20.00 = 2 x $10.00
Parts total	$259.90
Total	$23,451.10

The table above shows an estimate cost of development for this project. A main assumption made is that labor pay is the same as what an average graduate from ECE at Illinois makes, and that we work 8 hours everyday for a month. It is also noteworthy mentioning that this project will be funded by the MechSE department, with a budget of $1000 on top of the $40 given to us by the ECE department. We are also given an additional variable amount of money from the project sponsors based on need. Majority of the manufacturing cost will come from the electrical components relative to cost of housing/casing.

4 Ethics and Safety

Firstly, the sensor node that will be worn by an infant/toddler has to have safety features implemented into the design which aligns with the IEEE Code of Ethics #1. It states "to accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment." [13] For example, since there are bare electronics, we made sure non of it is in contact with the child by

having it in an enclosure. Also following the IEEE Code of Ethics, sources must be correctly cited and plagiarism must be avoided [13].

Since there is a microphone constantly recording, we have to align with the ACM Code of ethics "Respect the privacy of others." [14] To overcome privacy issues, all collected and streamed data will be hardware and software encrypted and only authorized users will be able to decrypt and use said data/information. Most of the briefly mentioned hardware components above do have built-in hardware encryption accelerators and software encryption is realized by encrypting each packet of data before transmitting. On the server side, encryption will be enabled for each step of the way. Also, all of the wireless communication components comply with the requirements by the FCC.

To protect the rights and welfare of human subjects in research, we would have to abide to the Institutional Review Board (IRB) which takes oversight of research involving human subjects. [15][16] With the research coordinators leading the IRB review process, we will have all the required approval in order to carry out testing with proper consent and in accordance to IRB protocol. Also, AWS has never been used in a research project before, though we managed to get full approval from the IRB with the help of Professor Johnson.

5 Conclusion

With all the progress we have made so far, we can definitely call this project a success. Most of our initial requirements are met and verified. That being said, this was planned to be a year long project, so there is still more room for improvement. In the following semester, we plan to improve on a few things: make ECG signals less prone to noises from movement, apply digital filters on microphone recordings, have higher capacity rechargeable batteries (nickel metal hydride batteries) and redesign the sensor node PCB such that it can fit into small pockets sewn onto an infant's vest. Overall, this has been a great learning opportunity for all of us and we are proud to call this project our own.

References

[1] McElwain, N. L., Pomerantz, E.M., Bub, K. L., Hahn, L. J. W., Johnson, H. T., & Bernhard, J (2017) Children in the Wild: Engineering Tools to Capture Child Development in Real-World Contexts, SBSRI Small Grants Program

[2] Porges, S. W. (2007). The polyvagal perspective. Biological Psychology, 74(2):116–143.

[3] Porges, S. W. (2011). polyvagal theory: Neurophysiological foundations of emotions, attachment, communication, and self-regulation. New York, NY: Norton.

[4] Bub, K. L., Buckhalt, J. A., & El-Sheikh, M. (2011). Children's sleep and cognitive performance: a cross-domain analysis of change over time. Developmental psychology, 47(6), 1504.

[5] Cole, J., Berry, D., McElwain, N. L., Bulkes, N., Mahrt, T., & Emery, H. T. (June 2015). Linking pitch prosody and affective behavior in mother-child interaction. Paper presented at the Workshop on Prosodic Development at the International Biennial Conference on Phonetics and Phonology. Cambridge, United Kingdom.

[6] McElwain, N. L., Berry, D., & Cole, J. (May 2016). The sound of sensitivity: In-the-moment relations between maternal pitch prosody and mother-toddler interaction quality. In A. Bernier (Chair), Recent advances in the study of early maternal behavior. Paper symposium to be conducted at the XXth Biennial International Conference on Infant Studies, New Orleans, Louisiana.

[7] Espressif Systems (October 2017). ESP32 Datasheet v1.9. Available: http://espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

[8] MEMS microphones vs Electret microphones [Online]. Available: https://ez.analog.com/thread/11410

[9] ESP32 WiFi & BLE module [Online]. Available: https://www.seeedstudio.com/ESP-32S-Wifi-Bluetooth-Combo-Module-p-2706.html

[10] GATT Specifications [Online]. Available: https://www.bluetooth.com/specifications/gatt/

[11] Analog Devices single lead, heart rate monitor front-end [Online]. Available: http://www.analog.com/media/en/technical-documentation/data-sheets/AD8232.pdf

[12] MQTT [Online]. Available: http://mqtt.org/

[13] IEEE Code of Ethics [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html

[14] ACM Code of Ethics [Online]. Available: https://www.acm.org/about-acm/acm-code-of-ethics-and-professional-conduct

[15] Urbana-Champaign Policy Governing the Use of Human Subjects in Research [Online]. Available: http://cam.illinois.edu/xi/xi-1.htm

[16] Review Processes and Checklists [Online]. Available: https://oprs.research.illinois.edu/review-processes-checklists

[17] ECG Measurement [Online]. Available:
http://www.cisl.columbia.edu/kinget_group/student_projects/ECG/Report/E6001/ECG/final/report.htm

[18] Digital Sampling Rate and ECG Analysis [Online]. Available:
http://www.sciencedirect.com/science/article/pii/0141542585900275

[19] SPH0645LM4H Datasheet [Online]. Available:
https://cdn-shop.adafruit.com/product-files/3421/i2S+Datasheet.PDF

[20] ESP32 WROOM schematic [Online]. Available:
https://dl.espressif.com/dl/schematics/ESP-WROOM-32-v3.2_sch.pdf

[21] LM2596 Datasheet [Online]. Available:
http://www.ti.com/lit/ds/symlink/lm2596.pdf

[22] LM2596 Datasheet [Online]. Available:
http://data.energizer.com/pdfs/e92.pdf

[23] Average ECE salary [Online]. Available:
https://ece.illinois.edu/admissions/why-ece/salary-averages.asp

6 Appendix

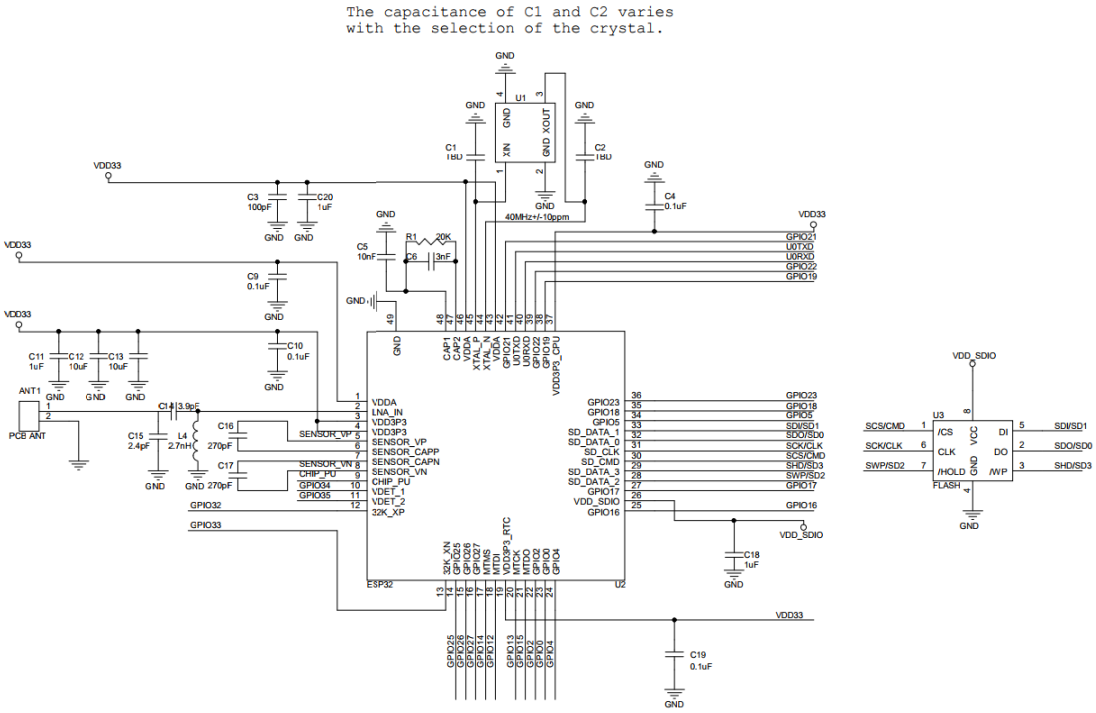

Figure 30: ESP32 WROOM schematics. [20]

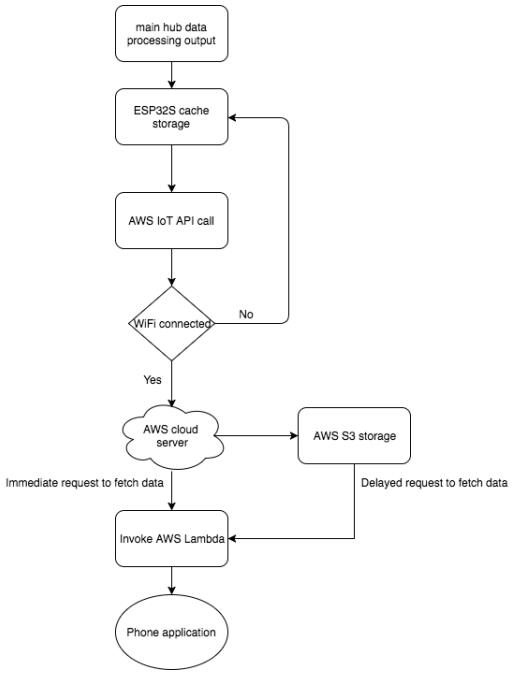

Figure 31: Flowchart for AWS data back-end.

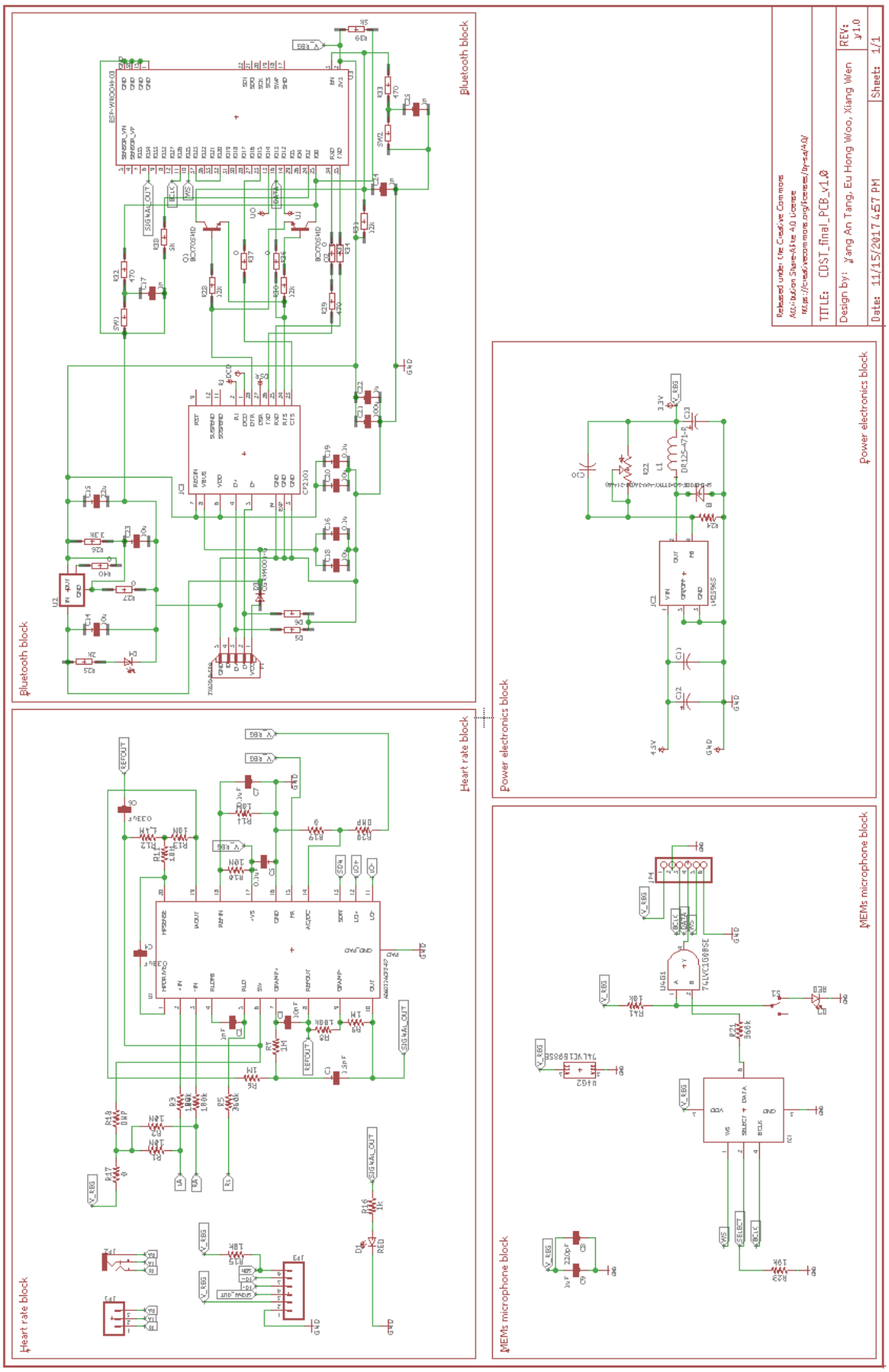

Figure 29: Sensor Node Schematic

Table 6: Sensor Node - Physical R&V Table

Requirement	Verification
Overall dimensions of the final PCB with components mounted should fit the dimensions specified by the mechanical engineering team.	**Final Dimensions** 1. Final PCB with components mounted should fit specified dimensions. PCB will be placed into housing to make sure it conforms to dimensions.
Power consumption should be low enough to keep the temperature of the housing/casing to below 100 degrees Fahrenheit (37 Degrees Celsius).	**Temperature** 1. A thermometer gun will be used to monitor under all typical operations during development.
Needs to be light so that it does not affect an infants movements. Sensor node ideally around 1.5 lbs (± 0.5 lbs) excluding housing/casing.	**Weight** 1. Weigh the whole PCB with batteries and all the components soldered on using a scale.
Does not have loose parts that can easily be ingested by an infant or toddler.	**Safety** 1. Final product will be worn by child through multiple use cases to affirm no moving parts.

Table 6: Wireless R&V Table

Requirement	Verification
Have the data collected compiled and converted into a readable file format available for playback.	**Data Format** 1. Timestamps will be recorded for ECG data before packets are uploaded for easier playback.

Table 6: Sensor Node - Electrical R&V Table

Requirement	Verification
ECG sensors will need to be at least 70% accurate relative to the waveform obtained from a typical ECG sensor used by the research team. Or qualitatively, out of the 5 main dips and peaks observable from an ECG waveform, only the R peaks needs to be easily distinguishable from each other.	**ECG Waveform** 1. ECG waveforms obtained from our prototype will be compared with the current ECG sensors used by the research group. 2. ECG leads from our prototype and the one used by the research team will be attached to the same person at the same time. The mean absolute percentage error of the RR-I's can be computed, and verified to be lesser than 30%.
Voice recordings need to be sensitive and clear enough such that there is audible speech obtained from the recordings. Preferably, 16-bit mono samples at around 5 kHz sampling rate, and a signal to noise ratio of about 50 dB-A ('A' means weighted at 1kHz signal)	**Voice Recording** 1. This can be verified by testing if a recording of someone reading a passage can be discerned when played back. 2. Using MatLAB FFT algorithms, compare the frequency spectrum of source recording with microphone output to deduce SNR.
Voice recordings might invade a user's privacy, so there must be a physical button to allow users to stop the recordings at any time.	**Hardware Mute** 1. Verify that when the button is pressed, microphone is disabled and no audio data is received by main hub. This can be done by constantly reading serial output of main hub.

Table 6: Bluetooth R&V Table

Requirement	Verification
Bluetooth ranges can reach at least 20 feet so that an infant is able move freely within a room.	**Range** 1. Verify that it works at a distance of 10 meters by having the sensor node placed exactly 20 feet from the hub measured using tape measure. 2. The signal strength at 10 meters should at least be around -70 dBm measured.

Table 6: Buck Converter R&V Table

Requirement	Verification
Design a DC-DC buck converter using a 4.5 V AAA battery as source, supplying 3.3 V to the rest of the circuit.	**Supply of power** 1. Input voltage waveforms to components will be obtained from the oscilloscope. 2. Operation of individual components as load for all operation will be tested separately to verify functionality of power supply. (a) Microphone voice recording while powered. (b) ECG chip data read while powered. (c) Bluetooth module data transmission while powered. 3. All components will be powered simultaneously and all functions will be tested.
Voltage waveform must have +/- 0.3 V ripple from 3.3 V.	**Consistent Output Voltage** 1. Output voltage waveform of rectifier will be probed with an oscilloscope under all operation. 2. Ripple will be measured using measurement tools, such as cursors and average function, in the scope.
Operation time of the batteries must be at least one research session or equivalently 3 to 4 hours.	**Operation Time** 1. Batteries will be stress tested at equivalent maximum load until the converter shuts off. (a) Battery will be connected to a 30 Ω resistor. (b) For each 10 minute interval, voltage on the battery will be measured using a multimeter before and after.

Table 6: Rectifier R&V Table

Requirement	**Verification**
Design an AC-DC full-wave rectifier using the wall outlet as source and must supply 3.3 V to both Bluetooth modules and 5.0 V to the processing unit. All components must have the required power to function for all operations.	**Supply of power** 1. Input voltage waveforms to components will be obtained from the oscilloscope. 2. Operation of individual components as load for all operation will be tested separately to verify functionality of power supply. (a) Processing unit and all functions will be tested (i.e. data transmission via WiFi) while powered (b) Bluetooth module and all functions will be tested while powered 3. All components will be powered simultaneously and all functions will be tested.
Voltage waveform must have +/- 0.3 V ripple from 3.3 V for the Bluetooth modules and 5 V +/- 0.25 V for the processing unit. Operation of components fail above and below the ripple values and therefore ripple must constrained.	**Consistent Output Voltage** 1. Output voltage waveform of rectifier will be probed with an oscilloscope under all operation. 2. Ripple will be measured using measurement tools, such as cursors and average function, in the scope.