Title
Wireless Advances in the Monitoring of the Human Gastrointestinal Tract

Author
Wasko, Michael Hong Lee

Publication Date
2015

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Wireless Advances in the Monitoring of the Human Gastrointestinal Tract

A thesis submitted in partial satisfaction

of the requirements for the degree Master of Science

in Electrical Engineering

by

Michael Hong Lee Wasko

2015

ABSTRACT OF THE THESIS

Wireless Advances in the Monitoring of the Human Gastrointestinal Tract

by

Michael Hong Lee Wasko

Master of Science in Electrical Engineering

University of California, Los Angeles, 2015

Professor William J. Kaiser, Chair

Significant transducer, embedded, and wireless advancements on a novel system, *AbStats*, has led to the first non-hospital commercialization of a new system called, *Wireless AbStats*. The architecture for the *Wireless AbStats* system is presented. This thesis will cover the hardware and software design and implementation of the system, which includes the transducer properties of the acoustic sensor, an embedded node for local storage and wireless communication, a powerful, intermediate smartphone node for data collection, and a server enterprise for further integration with internet services. Additionally, we see improvements made to various aspects of the system, like design, manufacture, and deployment since the previous iteration. Overall, the new capabilities offered by the *Wireless AbStats* system, which include wireless support, smaller form-factor, and improved signal-to-noise ratio, can lead to new insights and solutions in high-impact and far-reaching problems and applications, such as veterinary care, animal husbandry, neonatal intensive care unit monitoring, and optimal athlete nutrition.

The thesis of Michael Hong Lee Wasko is approved.

Robert N. Candler

Gregory J. Pottie

William J. Kaiser, Committee Chair

University of California, Los Angeles

2015

Table of Contents

Acknowledgements

First of all, I'd like to thank my committee for accepting my thesis on such short notice.

I'd also like to thank the members of the UCLA Wireless Health Institute Lab (WHI) for sharing their knowledge with me, introducing to me different platforms and technologies, as well as giving me inspirations for new solutions.

I'd like to thank Vince Zegarski, a member of WHI, for sharing his knowledge on *AbStats* with me, as well as acclimating me with the details of the *AbStats* project so that I was able to make significant advances with it.

I'd like to give a special thanks to Professor Kaiser for giving me the opportunity to 2 years ago to join the Wireless Health Institute, especially when I was at a point where I wasn't sure what direction to head in career-wise. My credentials at that time were not optimal, yet he still took a chance on me, and it has made all the difference. All of the lab opportunities and experiences which ranged from product development insights to business-related considerations have truly been enriching and could have never been obtained in a class.

Lastly, I'd like to give a loving thanks to my parents for keeping me on track and supporting me when I seemed to get a little lost. Without their undying love, support, and empathy, could I have never finished this thesis.

Chapter 1. Introduction and Recap on *AbStats*

1.1 Abdominal Surgery and Consequences of Extended Length of Stays

Following abdominal surgery, patients typically suffer from restriction in bowel function which is characterized as a condition known as postoperative ileus (POI). Upwards of one in four patients who undergo surgeries in the abdominal region—for example, a colectomy, which is the removal of a portion of the large intestine (colon)—experience some form of POI [1]. One important with problem with a condition like POI is that there spawns a number of complications, including an increase in the patient's length of stay (LOS) in the hospital and overall discomfort experienced by the patient [2]. With POI, patients may experience nausea and vomiting, possibly leading to aspiration pneumonia which is caused by breathing liquid or food into the lungs. Nationwide, abdominal surgeries result in 1.8 million hospital days and cost a total of $1.46 billion annually, as seen below in Figure 1, where patients with POI have a 30% longer LOS than those without. Reducing the average LOS can save in medical costs as well as help to clear up hospital resources. This is particularly important because this can affect a hospital reputation and department funding within the hospital. Additionally, space required by the hospital may also be in demand, so an increase in LOS can prove problematic.

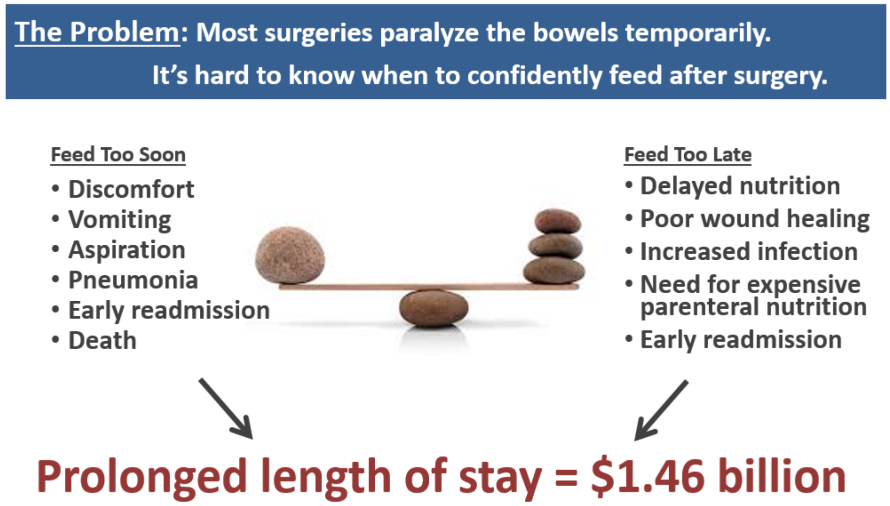

Figure 1: Cost of Prolonged Length of Stay due to Post-Operative Feeding Recovery

The human gastrointestinal (GI) tract is shown below in Figure 2. Bowel function can be categorized as the movement of chyme, which are semi-solid masses that are expelled by the stomach into the duodenum [3]. Bowel sounds are a widely accepted indicator of bowel function, as well as proceeding bowel movements or passing of flatus. Bowel motility is performed by a mechanism called peristalsis, the powerful contraction and relaxation of the smooth bowel muscles to push contents through the gastrointestinal tract. These peristalsis events are the primary source of the sounds that characterize digestion and thus can aid in the verification of proper bowel function. The converse is also true: the absence of these events can help in the detection of ileus.

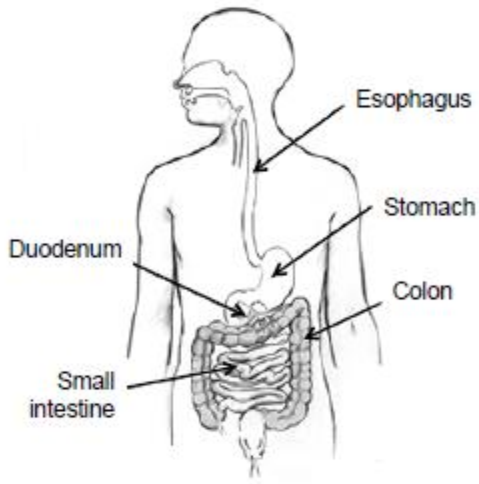

Figure 2: The Human Gastrointestinal Tract [4]

After abdominal surgery, hospitals can choose to prohibit the patient from ingesting any solids or liquids, *nil per os* (NPO), for the first and second postoperative days (POD). The transition consists of the NPO, then to a liquid diet, and then to a solid food diet. While "fast-tract" schemes allow liquids on POD zero as soon as the patient awakes from anesthesia, this plan may still be too aggressive. As a result, POI can lead to many of the complications, which results in returning the patient to the NPO nutrition delivery. On the other hand, if POI is not present, the transition to a solid diet may be too slow and the patient may not be able to receive proper nutrition and can experience discomfort when NPO is extended. Essentially, the only way to plan the diet is by observing the bowel function, which has been done in the past by using a stethoscope to listen to the peristalsis events and determining the proper course of action. However, this method in itself is insufficient as the right plan only becomes more apparent with longer, accurate listening periods – something that is physically not feasible due to precious physician time. Depending on the level of bowel functionality, peristalsis events may not occur frequently enough to be heard in short examinations, which can lead to an inaccurate estimate of current digestive health. As a result, diets are given, but are only changed when severe problems become apparent. *AbStats* has provided a solution to this problem by providing quality temporal acoustic recordings of digestive health, and semi-automating the whole process by detecting the essential peristalsis events.

The events of the migrating motor complex (MMC) are composed of a system-wide series of periodic and less-powerful events to help “clean out” the GI tract. There has been extensive research on the MMC from [5], [6], and [7], but it is with the development of *AbStats* that has autonomously indicated when and how frequently these events occur.

This thesis will outline the latest *Wireless AbStats* system, including the embedded hardware used to record of the abdominal sounds, the flow of data, and potential extensions of the technology in emerging applications. The presented design combined with the emerging applications will change the way people will monitor not only digestive, but bioacoustics sounds worldwide. However first, we recap on the existing components to the original *AbStats* design.

1.2 Disposable Sensors

The inspiration for the *AbStats* acoustic sensor came from stethoscopes that are still used by physicians in hospitals today, like that shown below in Figure 3.

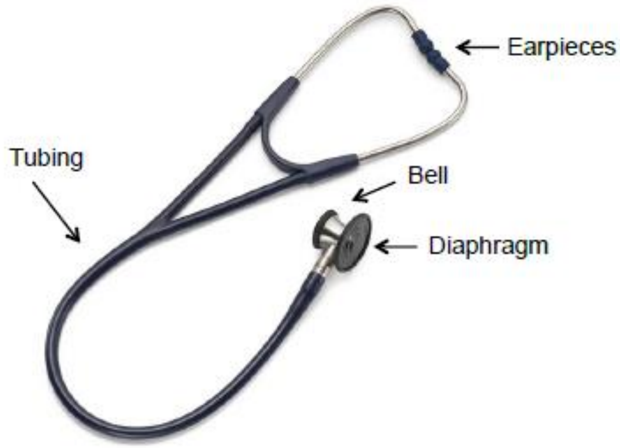

Figure 3: Welch Allyn Harvey Stethoscope [8]

Sounds are mechanically coupled from the diaphragm of the stethoscope to the air inside the chamber of the bell. A similar, but enhanced approach is taken with the *AbStats* sensor. In the case of *AbStats* an electret microphone lies inside the chamber and the diaphragm is made out of an extremely flexible, high compliance material. We currently use an extremely-low durometer silicone material. As a result, mechanical reverberations are felt by the diaphragm and modulate the pressure within the chamber. Next, the electret mic in the chamber responds to pressure changes in the chamber instead of to wave vibrations hitting the electret mic, as it transduction was originally intended. The result is extremely favorable. Here, the responsivity of the device is a few order of magnitudes higher than the original stethoscope, as well as the increased noise and interference rejection improvements. By taking the existing design of the stethoscope, understanding the transduction principles, and iterating many times on designs, a modest, but effective replacement has been developed.

In Figure 4, a side-view and top-view of the original *AbStats* sensor is shown. In this design, a plastic shim was connected to a black ABS housing. These shims were then glued to *Tegaderm* transparent-film dressing (bandages). This allowed proper adhering to the patient's abdomen that is strong enough to hold the disposable sensor in place, while still maintaining comfort for extended periods of time.

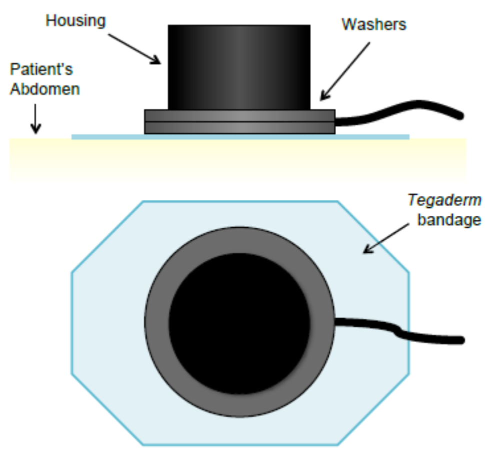

Figure 4: Diagram of a Disposable Sensor. Side View (top) and Ariel View (bottom)

Small coin-cell vibrating motors are added to every sensor to equip them for coupling tests. These do not contribute to the bioacoustics sensing and transduction mechanisms, however. Inside the sensor housing, a circular prototyping board is used to hold the electric microphone and vibrating motor. This board allows easy manufacture of the sensor by soldering the motor and electret mic to the board. Additionally, it is a robust method for housing electronic components.

1.3 Electret Microphone

For *AbStats* and other bioacoustics applications, the electret microphone used in the disposable sensor must be capable of capturing most of the audible frequency range. This typically ranges from 20 Hz to 20 kHz of the audio range. One of the major reasons, particularly for *AbStats* to further support this electret mic choices is because sounds emitted during peristalsis are in the low frequencies, approximately between 80Hz and 200Hz. Given these frequency range considerations, the two important criteria that needed to be met was the electret that was chosen provided a useful frequency response and that it was very inexpensive.

The 6mm diameter, 3.5mm tall electret microphone that was chosen to be inside the plastic housing of the disposable sensor is the XCM6035 from SPL, pictured below in Figure 5. There are two pads on the back of the microphone which are soldered to the PCB. The signal collected by the electret microphone is then sent to the circuit near the biasing and amplifying network, which will be explained later in the paper.

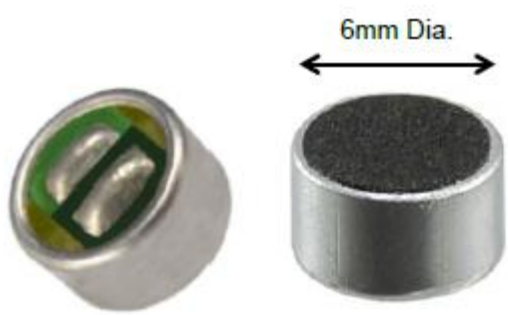

Figure 5: Photo of the XCM6035 Electret Microphone. Back (left) and Face (right)

This electret microphone provides an output frequency response that covers most of the audible range, as seen below in Figure 6 [9]. This frequency response is attractive, as it provides a nearly flatband response for the frequencies of bioacoustics interest. This type of raw signal response will be useful further down the signal pipeline when signal processing algorithms are implemented.

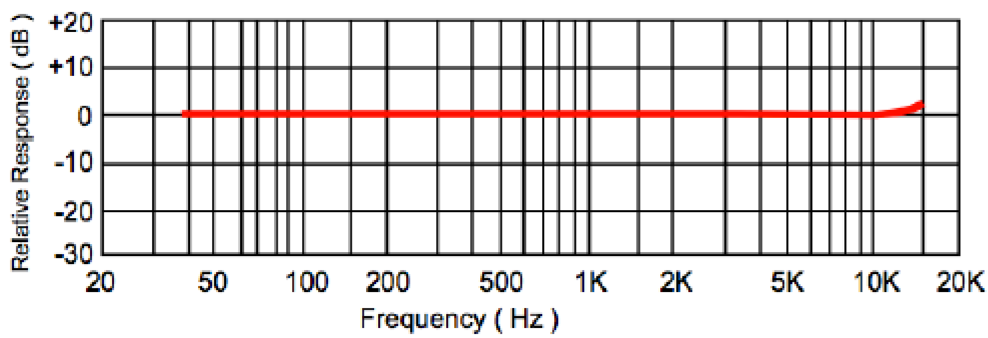

Figure 6: Frequency Response of XCM6035 Electret Condenser Microphone

1.4 *Wireless AbStats* End-to-End System Overview

Next we present a block diagram to summarize essential components to the *Wireless AbStats* system. The system can be decomposed into three essential nodes. These nodes contain dedicated hardware that is serves an essential function in the data pipeline and is implemented appropriately based on its electronic hardware capabilities and form-factor. The first node is the combination of the acoustic sensor, which has been modified and advanced since the original *AbStats* and the embedded node that it interfaces to. The second node can exist as many different types of devices, which include smartphones, tablets, laptops, and even the legacy *AbStats* gateway which is being upgraded to support more wireless data transport protocols. However, for the majority of the paper, we refer to the intermediate node as a smartphone as the software that was implemented was done on one. The last node is the server enterprise which can run a variety of different software platforms, including Linux shell scripts, PHP scripts, and algorithms, as well as host webpages and store data files. Data is transported from the embedded node to an intermediate node via Bluetooth. For the purposes of this thesis we implemented this using the Bluetooth 2.0 standard, however, at the time of writing this thesis, there are current efforts to implement Bluetooth Low-Energy (BLE) which was introduced with the Bluetooth 4.0 standard. The data is transported from the intermediate node to the sever enterprise using Wi-Fi or a cellular network which are commonly 3G, 4G, and LTE networks.

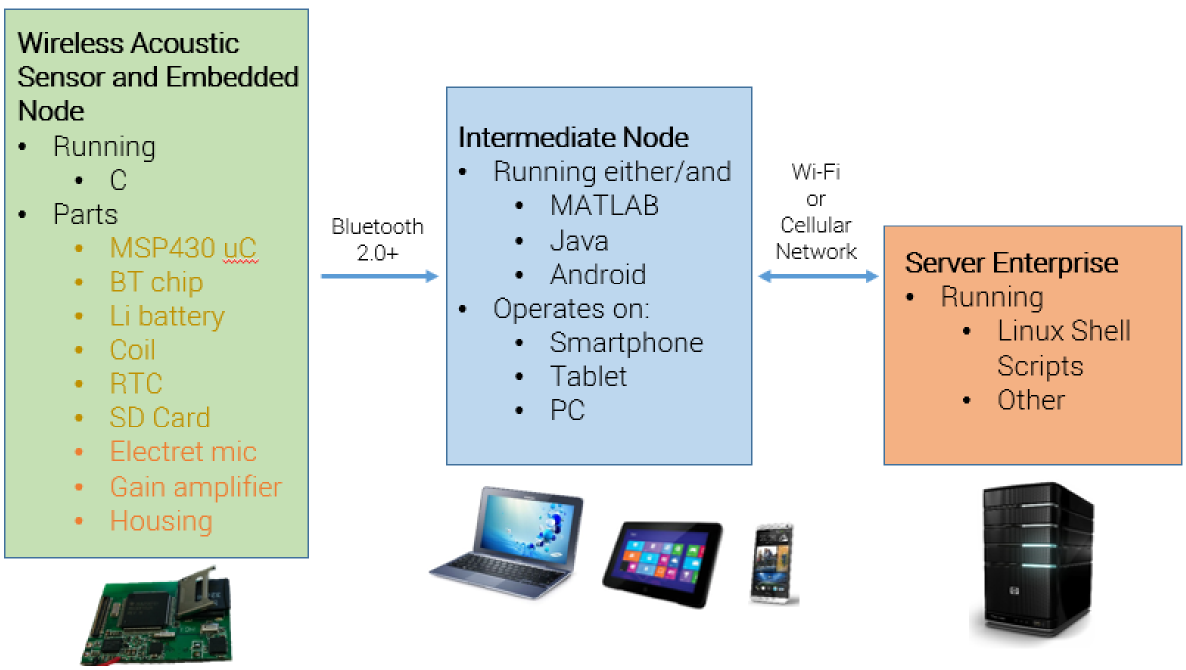

Figure 7: *Wireless AbStats* System Block Diagram

Chapter 2. Improvements to the sensor transducer technology.

2.1 Improvement of signal to noise ratio via complete sealing of sensor housing

In the previous version of *AbStats*, sensor packaging aimed to sense pneumatic changes in the housing chamber due to contractions of the skin on the membrane wall of the sensor housing. In principle, this is scientifically sound, but wasn't completely realized. The 3.5mm audio jack in the sensor housing actually created a small opening that disengaged the pneumatic principals. By sealing the rim of the 3.5 mm jack with industrial grade adhesive, the seal was complete and pneumatic principals were strongly engaged in the transduction process. This in itself has opened a new dimension to biological acoustic sensing, and will continue to produce a plethora of applications, for example, in heart and respiratory monitoring which are not the subject of this paper. To elaborate, heart sounds as measured by *AbStats* was on the same signal-to-noise ratio as common stethoscopes, however, by completely sealing the sensor, acoustic signal detail has appeared that has never been seen before, as evidenced below by Figure 8.

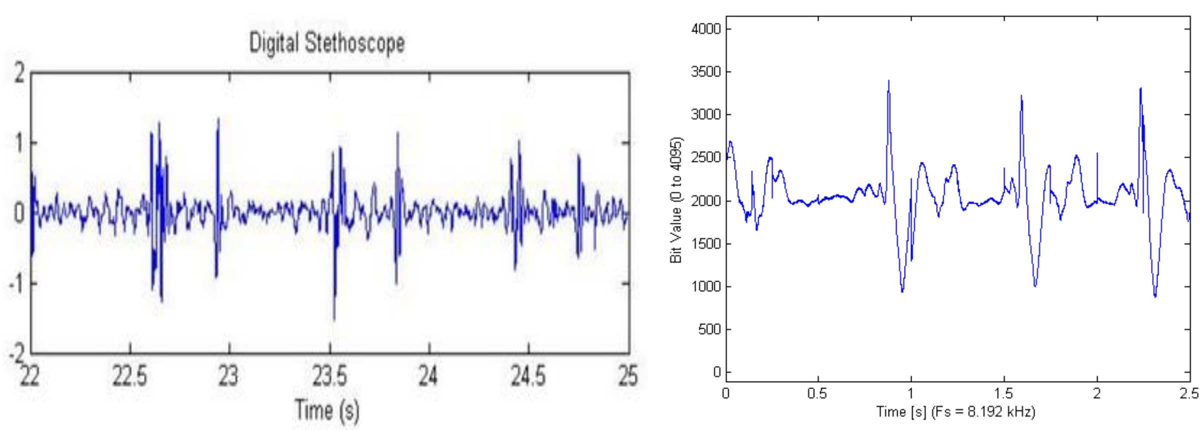

Figure 8: Left -Typical Digital Stethoscope Signal Recording, Right - Improved *AbStats* Acoustic Sensor

2.2 3D-printed housings and its effect on cost the manufacturing pipeline.

The original housing design consisted of combining a plastic shim with an ABS cylindrical housing. However, because these parts are separate, there exists a process of manufacture which was completed via human man-hours. Additionally, quality control was also an issue in this process and led to the some sensors becoming defective before even being employed or shortly after deployment. For example, poor adhesion of the shim to the cylindrical housing was yet another culprit in poor housing seal, significantly reducing signal quality. Wasted monetary and time resources are further amplified as the whole *AbStats* system has no way of detecting malfunctions in sensor housings as the signal captured does not generate feedback that can alert of these errors.

To combat these issues, resources were invested in 3D-printing technology to manufacture the housings. In-lab prototyping was done using a commercial 3D printer called the *MakerBot.* Once the design was finalized, it was sent to Solid Concepts, a *Stratasys* Company. The technology they used to print the sensors was Selective Laser Sintering (SLS). While the general cylindrical shape of the housing has not changed, the quality control and robustness of the system has improved significantly. Additionally, there is no human man-hour overhead to producing the sensors.

Figure 9: Selective Laser Sintering 3D Printed Housing by Solid Concepts

Further iterations of this process led to the outsourcing of the 3D-printing process, which, as a result, led to higher quality prints and saved man-hours on our team's end.

2.3 Application dependent sensor membrane variations

The previous iteration of *AbStats* employed the use of Tegaderm, a disposable, film-like bandage. Because *AbStats* was originally designed for the hospital setting, the deployed devices needed to be disposable. In the case of *Wireless AbStats*, which is targeted to the commercial home-use market, the disposable characteristic of the sensor system is no longer a requirement.

To address this new design consideration, red silicone of the lowest durometer was deployed to the open, exposed end of the housing and sealed with adhesive, forming a flexible gasket. The low durometer means that the compliance of the material is extremely high, and can conform to the bodily surface that the sensor housing is applied to. At this point, there are a couple of methods to applying the sensor to the body. The first is like the traditional stethoscope method, in that one can simply hold the sensor with the red silicone membrane against the subject's body. The other method is by using an elastic strap to clamp the sensor onto the subject's body.

Additionally, another sensor application method was used and is similar to the Tegaderm method. Here, we cut out a ring of 3M VHB tape with an outer and inner diameter matching that of the sensor housing rim. This VHB tape ring is applied to the rim. Then, an ECG electrode with adhesive by 3M was modified, by punching out a hole with a size that matched the inner diameter of the sensor housing. This got rid of the AgCl electrode in the process. This ECG adhesive is then attached to the VHB tape, adding to the housing stack. When data needs to be collected, the housing stack is attached to the subject and data collection can proceed. One advantage of this method over the Tegaderm method is that it is more immune to skin contraction and expansion. Additionally, the adhesion that occurs between the skin and the sensor stack is much stronger and can be deployed for long periods of time, reliably. However, both Tegaderm and ECG adhesive methods are single use and require fresh materials for sensor redeployment.

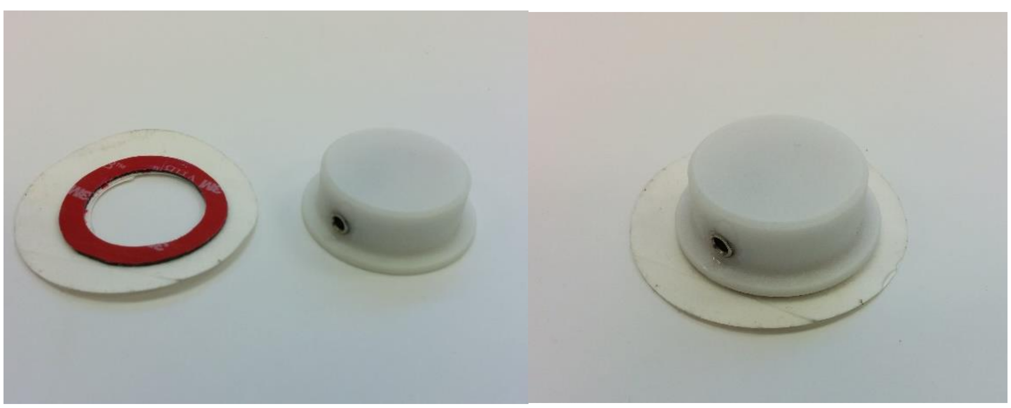

Figure 10: 3M VHB Tape and Modified ECG Electrode Adhesive Sensor Combination Stack

Chapter 3. Design of the embedded system node

3.1 Product packaging and field deployment

The original design of the *AbStats* sensor involved a few required constraints as a result of the hospital application. One major one was that the node placed on the body needed to be disposable. This is because of certain sanitation issues that would make passing this device for hospital use much more difficult. Additionally, since each sensor node needed to be disposable, an indirect requirement was that the sensor nodes needed to be relatively low-cost.

However, this design factor has changed for the situation of *Wireless AbStats*. If one wanted to design and market a wireless health monitoring device, the market operating space needs to be carefully considered. Entities such as CE Mark and the Food and Drug Administration can slow the certification of a device significantly if it is to be used for medical diagnoses in a hospital settings. Thus, *Wireless Abstats*, although it is a technological advance in wireless healthcare, its current design is currently only compatible with the non-hospital commercial market for the reasons that will be explained next.

The design consists of a 3.5mm audio cable attached to the cylindrical acoustic sensor housing. However, this 3.5mm audio cable has been shortened and doesn't connect to an external gateway like in the previous *AbStats*. Instead, the connections of the cable are connected to the embedded system node which contains many non-disposable components (from a cost perspective). The parts will be explained in detail in the next section. The 3.5mm audio cable's connections have mechanical strain reliefs which consist of hot glue and other adhesives which are applied to extra real-estate on the embedded system node's printed circuit board. A groove to let the 3.5mm cable into the housing and to the PCB was ground out using a Dremel tool. The resulting design is basically the acoustic sensor combined with the embedded system node via a 7 inch 3.5mm audio cable.

Figure 11: Embedded Node Connected to Acoustic Sensor

3.2 Modification of the acoustic sensor electronics

The electret microphone was originally biased through a 3.5mm cable with a 1 kΩ resistor and the signal was AC coupled via a 1 μF capacitor. However, for *Wireless AbStats*, this setup needed to be modified slightly. This is mainly because the ADC range of ADC12 module (which is explained in the next section) will read voltages between ground and a reference voltage (in this case the reference voltage is Vdd, which equals 3.3V). The coupling capacitor, however, will reject any DC biasing that occurs on the electret microphone side, and additionally, we need a way to bias the DC part of the signal halfway between Vdd and ground, to optimize for the full ADC range available.

All of this is achieved with a SparkFun Electret Microphone Breakout BOB-09964. This item provides the resistor bias and AC coupling capacitor present from the original design. Additionally, there is an op-amp on the breakout board, OPA344, and a resistive network that sets the DC bias halfway between the supply voltage and ground, and a gain factor of 100 for the AC input signal.

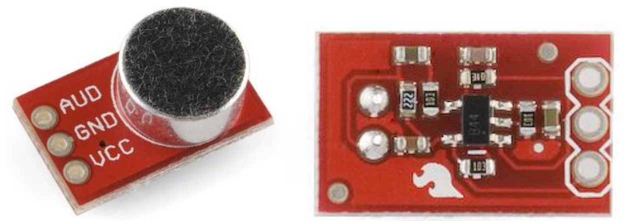

Figure 12: SparkFun Electret Mic and Inverting Amplifier Configuration Breakout [10]

However, the current state of the SparkFun breakout board out-of-the-box is not completely suitable for integration into *Wireless AbStats*. This is because the gain factor of 100 is simply too high. A typical raw heart sound as recorded by the electret mic is approximately 40mV peak-to-peak, so applying the gain will result in an output signal of 4V. All bioacoustics signals will thus "rail-out" by exceeding the 3.3V determined by the Vdd supply to the system. Thus, an effort to reduce the gain was taken. This was done by removing the 1 MΩ surface-mount resistor, as seen on the board and replacing it with a 35 kΩ one lowering the gain to 3.

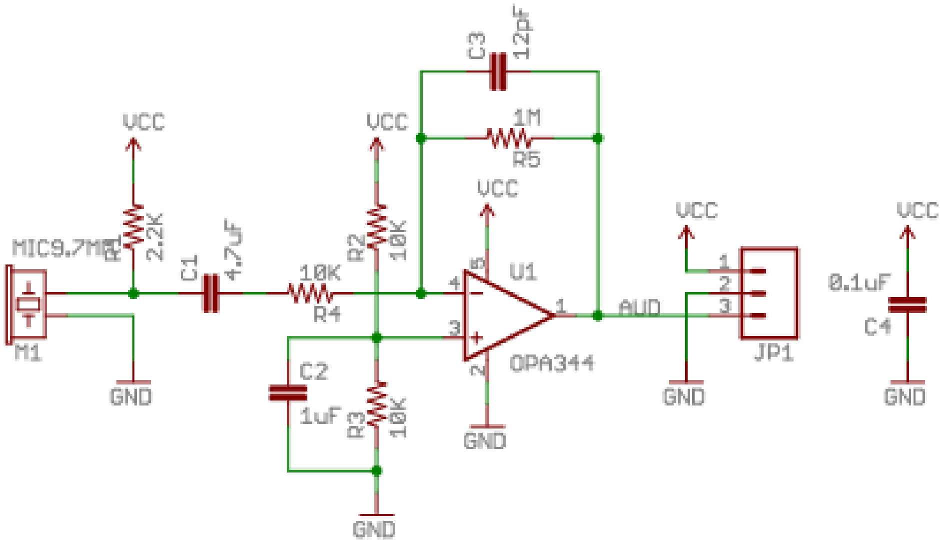

Figure 13: SparkFun Electret Mic and Inverting Amplifier Configuration Breakout Schematic [11]

Next, the analog circuitry from the electret microphone to amplifier output is explained. Resistor values refer to the schematic in Figure 13. R1 provides the bias to the electret mic. The signal is AC coupled through C1 and can be thought of as the input to the inverting amplifier configuration explained next. It is best to think about this amplifier configuration as what happens at AC and at DC. At DC, the capacitors act like open circuits, so the resistors R2 and R3 act as a voltage divider such that a DC bias voltage half of Vdd appears at the non-inverting terminal of the OPA344 op-amp. C1 is also treated like an open so, the voltage that appears at the inverting-terminal and the OPA344 output terminal is the same. Thus, at DC, the op-amp is acting as a unity buffer configuration where the output is half Vdd.

At AC, the capacitors can be thought of as short circuits. Since C1 is short, the AC signal passes from the electret mic into what will be apparent as an inverting amplifier configuration. C2 effectively shorts the non-inverting terminal to ground at AC. Since the input signal is fed towards the inverting terminal through R4, the configuration is exactly like that of the inverting op-amp configuration. Thus, the gain is set by the ratio of R5 to R4. As explained earlier, we replace R5 with a smaller resistor to lower the gain so that the output signal doesn't rail-out and instead lies in the voltage range between GND and Vdd. This superposition of the DC output voltage with the AC output voltage is the input to the ADC whose operation will be explained in the next sub-section.

3.3 The embedded system node

The embedded system node contains non-disposable electronic components that serve three main utility functions. The first is to sample the voltage signal from the acoustic sensor transducer using an ADC. The second function is to push the data via Bluetooth technology to a more capable node. The last function is to store data onto a local SD card in the event that the data cannot be pushed out instantly via BT. Figure 14 shows a block diagram of the components of the embedded node, as well as the communication interfaces used between the ICs on the board.

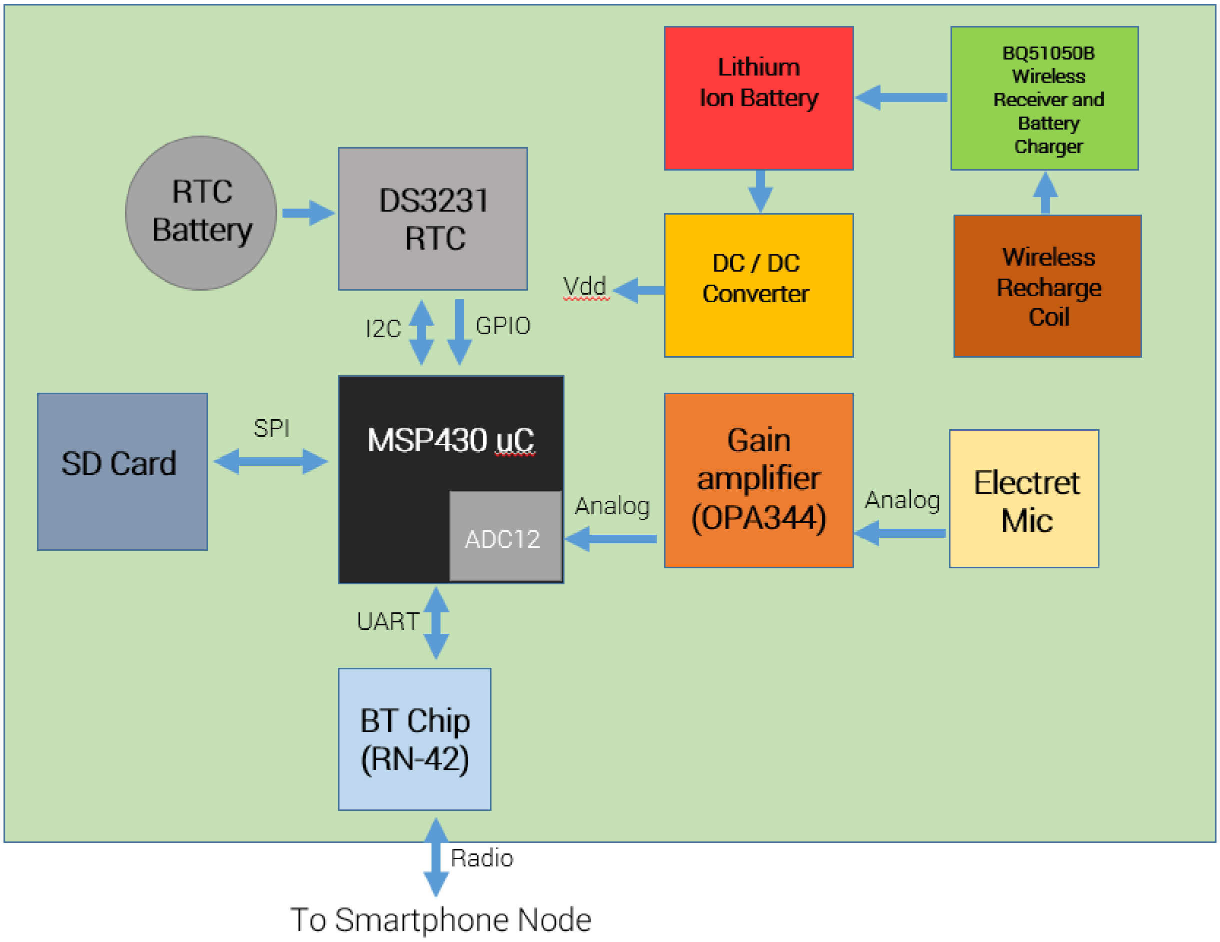

Figure 14: Block Diagram of the Embedded System Node and Sensor

Power supply: Battery, Charging Coil, and Charge management chip

The power management system consists of many parts: 5V Lithium ion polymer battery, wireless recharge coil, MAX17043 Fuel Gauge, LTC2955 Pushbutton Controller, LTC3536 DC/DC Converter, and the BQ51050B Wireless Recharge Receiver and Battery Charger. The operation is as follows: When the push-button is pressed, it shorts two GPIO pins to signal that the system should be powered on. The Lithium ion battery's typical voltage when charged is 5V. This voltage is the input to a DC/DC converter which steps the voltage down to 3.3V. This 3.3V is supplied to the rest of the system and is the system Vdd. When the push-button is held for a few seconds, this signals to the Pushbutton Controller to power down the system. When the embedded system is placed on a wireless recharge pad with the coil side adjacent to the pad, then the system will automatically recharge. This is done through the Wireless Recharge Receiver and Battery Charger. While the system can operate when the device is charging, the quality of the analog signal is extremely noisy and not acceptable for post-processing.

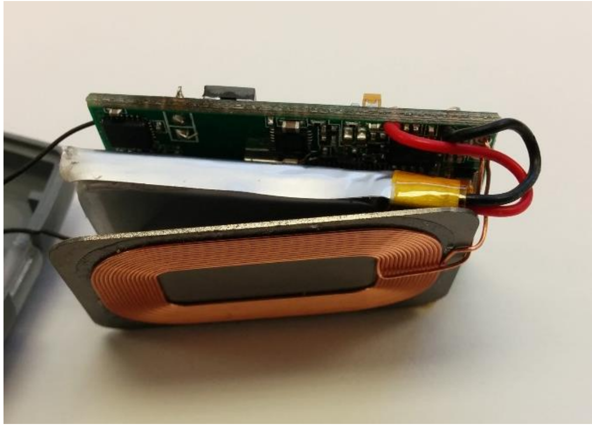

Figure 15: Embedded Node Stack from Bottom to Top: Charging Coil, Lithium Ion Battery, and Main PCB

MSP430 Microcontroller

The MSP430 microcontroller is a popular line of microcontrollers from Texas Instruments for low-power applications. It has many functions in our system. One function is for it to initialize the other ICs using chip communication protocols such as I2C and SPI. Specifically, the firmware is coded to enable certain sampling rates, clock speeds, as well as set certain chip configurations, all of which are done via I2C, UART, and SPI upon powering on the system. There are three important initializations that have to be made. The first is setting up the Bluetooth IC by setting the device name and baud rate so that the system can be recognized during pairing procedures with other nodes like smartphones, tablets and laptops. The second initialization that must be done is setting up the SPI module so that file writing to the SD card can occur. The next initialization involves setting up the ADC12 module on the MSP430 by configuring number of input channels and sampling methods. Another startup task is to check for certain system statuses, like battery life and if other system devices are operating as expected. There also other housekeeping initializations that must be done, some of which include, pin and port configuration, battery life reading, and variable and buffer initializations.

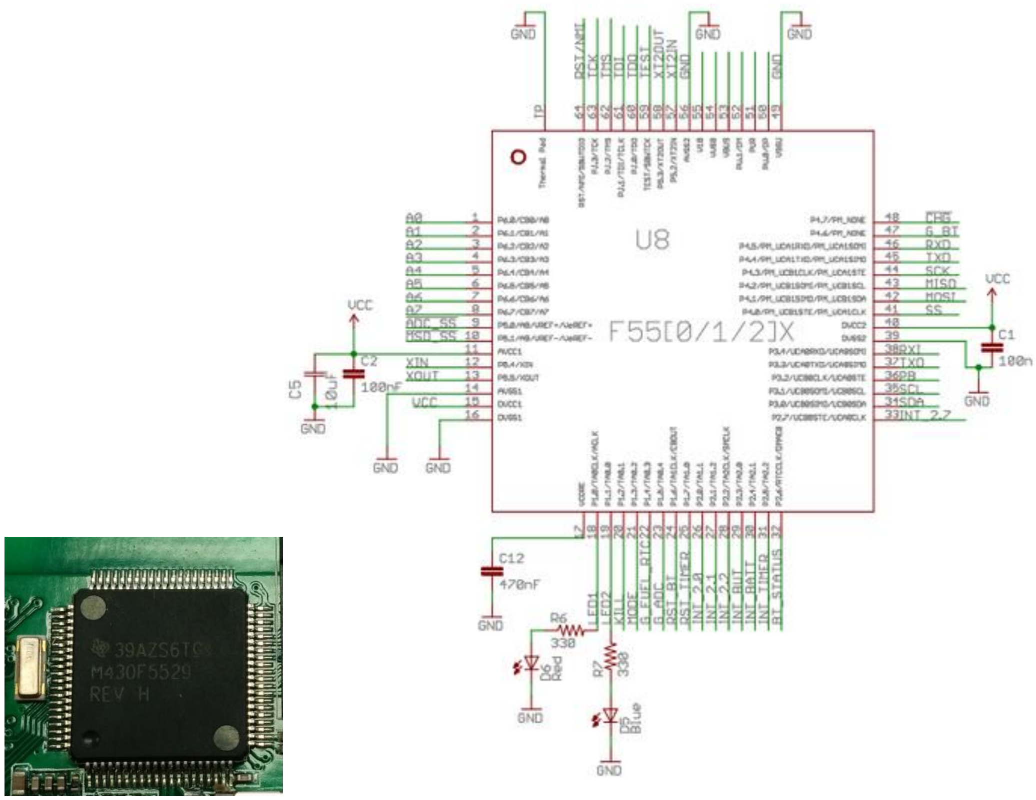

Figure 16: MSP430F5529 and Implementation Schematic [12]

Analog-to-Digital Converter (ADC) on the MSP430 (ADC12)

The original *AbStats* gateway leveraged the file standard of .WAV files, which included a 16-bit resolution for the signal. This design was fine because the operating system used utility functions which were all convenient in the development of the hospital gateway. For *Wireless AbStats*, we take the same design mentality of leveraging the features that are provided with the platform being used. The MSP430 solution already has an ADC module on the chip itself. However, the MSP430 doesn't provide an integrated 16-bit resolution ADC module. Instead, there are two ADC modules: the first being ADC10, a 10-bit resolution module, and ADC12, a 12-bit resolution module. While ADC12, naturally provides more resolution, the extra resolution may not be completely necessary for some applications. Thus, we design for performance first, and see how it affects other design factors, most importantly, battery life.

Real-Time Clock (RTC) Module

The Real-Time-Clock module for our embedded system consists of another separate IC: DS3231 Extremely Accurate I2C-Integrated RTC/TCXO/Crystal by Maxim Integrated. It communicates with the MSP430 chip via I2C protocol as well as a GPIO. Both communication interfaces are very important to how the system operates. The I2C communication is used to initialize the RTC. These initialization process include getting the current time from the chip, as well as setting the frequency of the square wave to be output on the GPIO. The GPIO pins on the RTC are wired to a programmable interrupt pin on the MSP430 and output a square wave at a frequency set during configuration. This square wave output only has four frequency options: 1 Hz, 1.024 kHz, 4.096 kHz, and 8.192 kHz. It was part of our design to use this square wave to sample the ADC. Not only would we have an accurate source of timekeeping, we would be able to provide it for the whole lifetime of the device and leverage these properties in the actual harvesting of the data.

Lastly, space has been allocated to the PCB to accommodate a coin-cell battery that can serve as a backup power supply for the RTC in the event that the main power from the Lithium Ion Battery and its regulating power system runs out or gets disrupted. This backup power feature would ensure that the device is still keeping the time if the application requires it.

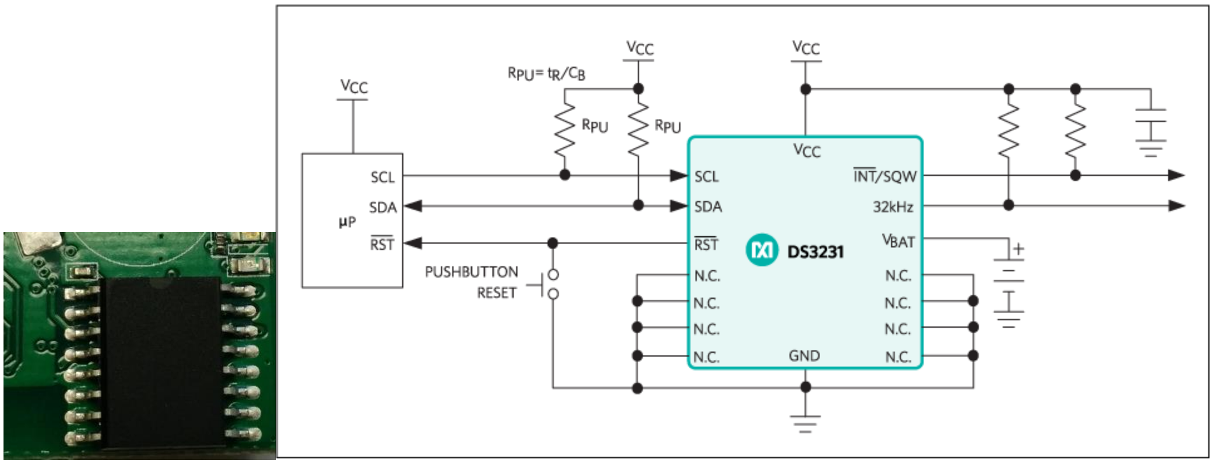

Figure 17: Maxim DS3231 Real-Time Clock Chip and Schematic [13]

Bluetooth Roving Networks 42 Chip

The Bluetooth chip that was implemented was the Roving Networks RN-42 chip. It communicates with the MSP430 microcontroller via UART. Data is passed from the MSP430 to the RN-42, as well as configuration settings. Additionally, data can be received wirelessly by the RN-42 chip to pass information to the MSP430 or even configure the RN-42 chip, however this receiving functionality is not used in the system.

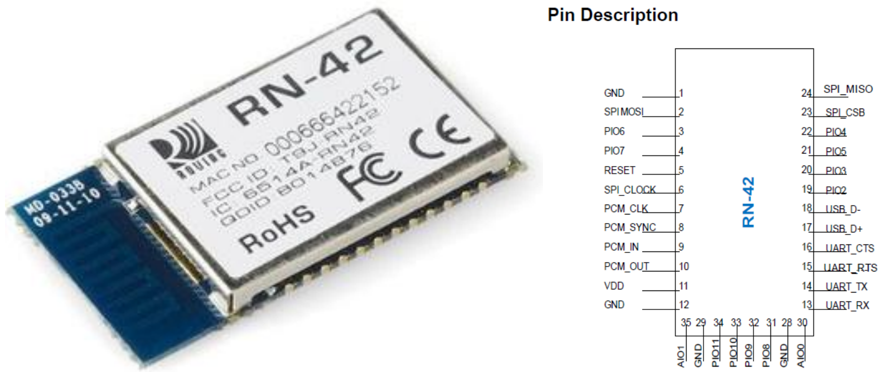

Figure 18: Roving Networks RN-42 Bluetooth Chip and Schematic [14]

SD Card

Figure 19 below shows the implementation of the SD Card and its connector on the embedded node PCB. It takes up more lateral space than the MSP430. The communication interface is SPI.

Figure 19: SD Card and Connector Implemented on Embedded System Node

Embedded Node Summary

The embedded system node design presented is a canonical design for wireless health applications. It features practical battery life, more than reasonable analog to digital conversion, small-form factor (with extra board real-estate), unused pins, and quality data transmission methods that can serve many applications. While some applications may have requirements that deviate from this design, this embedded design is a reasonable starting baseline and can easily be modified by adding and subtracting pieces as necessary, with minimal overhead.

Firmware Design Considerations

The firmware is written in C. Various libraries and drivers were supplied from Texas Instrument's website and were used and modified in the software design of the system. Other drivers and libraries were found for the other hardware chips and were used and modified accordingly.

There were a few design considerations in the firmware due to limited resources. These resources include ADC sample speed, buffer allocation size, and wireless output byte stream protocol. The first is the ADC sample speed. The final sampling rate is going to be application-dependent, but for digestive and heart monitoring applications, it was found that 4kHz was more than sufficient. While sampling rate can be easily picked for the ADC module, since it resides on the MSP430 chip, the next important thing that must be considered is the timestamp and drifting of the clock over long periods of time. This is particularly vital for the medical applications because data can be collected over the course of hours and must be accurate when saved to medical records. In order to solve this problem, we combine the interrupt feature of the MSP430 with the Real-Time Clock module chip. The RTC has a function of outputting a square wave that can be configured to 1 Hz, 1.024 kHz, 4.096 kHz, or 8.192 kHz. Further considerations are elaborated in the next paragraph regarding sampling rates generated by the RTC.

There is a tradeoff between the number of analog input channels used and the sampling rate. For *AbStats*, two channels are being used, but a closely-related application, Integrated Cardio Respiratory (ICR) uses four analog input channels. Also, while the ADC module of the MSP430 can operate at much higher frequencies, we want to practically eliminate clock drifting from our collected data. Thus, our design space is greatly reduced, but also simplified. So if we only consider four possible sampling rates, and up to four channels, we have to see if each combination is possible. The result is that for two analog input channels, it can be guaranteed that data can be sampled at 8.192 kHz by the RTC, pushed out via Bluetooth on time and not create a bottleneck. For four channels, it can be guaranteed that data can be sampled at 4.096 kHz by the RTC and, pushed out via Bluetooth, and not create a bottleneck. This sampling rate is sufficient for the ICR application as well. While there may be some intermediary value between the sampling rate of 4.096 kHz and 8.192 kHz for the each number of input channels, it is not completely necessary at this time, as the design space explained was able to meet the specifications of the applications with some margin for a safe design.

The next design consideration was buffer size. There is a little more than 5 MB available for buffer sizes after considering the size taken by other essential parts of the compiled code (drivers, functions, other variables). Thus, no matter what sampling frequency you choose, you have to divide this amount of memory to different buffers for each input channel. Additionally, when designing the system, the byte stream protocol that the system is designed around can also affect buffer size – for example, using a certain number of byte characters for delimiters.

The next design consideration was the wireless data protocol. While it was not the main bottleneck factor in the pipeline, this area had the most flexibility in terms of design, so the goal was to aim for a data stream protocol that was efficient, easy for software developers at receiving nodes to understand when developing their own software, and scalable in due to changes in sampling frequency and number of input channels used. Thus, byte protocol implemented is as follows:

Here, the timestamp is merely a counter that counts seconds, ultimately slaved to the RTC. Since we allocate four bytes of data to this counter, this time stamp can count more than four billion seconds. When the embedded node is powered on, the counter restarts from zero, so the counter is essentially counting the number of seconds since the device turns on. This allows the developer at the receiving end of the wireless byte stream to just understand that an extremely accurate counting timestamp is coming every second and they just need to keep track of how many counting timestamps came up since the first one received, which corresponds to the absolute, real-time start of the data collection.

Each sample of data for a given number of input channels is preceded by the two data header bytes and is followed by the actual data bytes. The number of data bytes is dependent on the number of input channels, but it is always a multiple of 2. This is because the ADC resolution is 12 bits, so we reserve 16 bits (2 bytes) for each sample of each input data channel. Thus a receiving node can be programmed to parse data based on how it is coming in. Either 2, 4, 6, or 8 bytes will follow the data header byte corresponding to 1, 2, 3, and 4 analog input channels. After that, the cycle repeats by outputting another data header byte if a new sample for a set of channels is ready, or a timestamp header byte and the timestamp if a new second of data is ready. While a receiving node can possibly infer the metadata, it is recommended that the developer of the receiving node already be aware of this metadata and program the receiving node to match the wireless data stream protocol. For this thesis, the programmer of the firmware and receiving nodes were the same so this was not an issue, whatsoever.

The same byte stream is saved to the SD card in order to efficiently use all of the SD card space. However, the only requirement if one wishes to read the SD card directly, they would still need to develop a parser in the same methodology that was required for the wireless receiving node. However, this development may not be completely necessary if pushing the data via Bluetooth can be scheduled for later; In this case, the byte data is simply stored to the SD card and pushed via Bluetooth at a later time, depending on the application.

Chapter 4. Interfacing with the Smartphone and server technology enterprises.

4.1 The smartphone data collector app

In general, the resources available to smartphones greatly exceed that of the embedded system node. Some reasons are the Linux kernel, a full operating system, multiple cores, relatively limitless memory, security, and much more capable radio interfaces. Thus, from a design perspective, we let the smartphone do most of the “heavy-lifting” when it comes to storing data, formatting the data into a human-readable file, and moving the data to a server enterprise. Another way to look at it is that the embedded node may need to be deployed for a whole day without interruption. This includes not being able to be charged until the day is over. The smartphone on the other hand, can exist in multiple places during deployment and data collection; it can be on a table, on or plugged into a charger, or even in a user's pocket. Since all of the information is digitized when communicating with the smartphone and pushed out on time, the burden of the embedded system node is appropriately reduced and in such a way that is consistent with the paradigm shift that we see taking place in the wearables market.

The smartphone that was used to test the data collector app was the Nexus 5 developed by Google and LG. The device contains a Snapdragon 800 system-on-a-chip by Qualcomm. The operating system during testing was Android 5.0. The data collection app was developed in Android Studio and the language is Android, which is extremely similar to Java in terms of operation and code syntax, however, with Android 5.0, the virtual machine executing the application has changed from the Java Virtual Machine (JVM) to Android Runtime (ART). While new APIs have appeared with Android 5.0's release, the APIs that we use to run the app are standard across all Android versions. Ultimately, we are limited by the hardware that the smartphone has to offer. This is an important consideration when ensuring compatibility across different devices, but it is not within the scope of this thesis. The good news is that the requirements to communicate with our embedded system are low and will continue to get lower as the advances in smartphone technology become ubiquitous and continue to hit the commercial smartphone market.

Figure 20: Nexus 5 Smartphone by Google and LG

The app is simple and has the following features: Automatic upload to data enterprise, manual upload to data enterprise, ability to choose the number of input channels used, and in-app tools to manage files such as clearing data or displaying it for preview. The automatic and manual data upload features are the main ones. The user must first pair the embedded node with the smartphone. This must be done only once, and can be done in the system’s stock settings user interface, or in the app. Next, the user should pick the number of acoustic channels that they wish to collect data from. It should correspond with what the firmware on the embedded system is configured to. The next step is to pick whether they want data to be uploaded automatically, or manually. For *AbStats*, since relatively long recordings are required due to transient nature and long sporadic spacing between peristalsis events, an automatic upload might be more appropriate to get the data quickly and consistently to the server enterprise. However, if the system is used in an experimental setting where the data collection sessions are only on the order of a few seconds to a few minutes, then, a manual upload may be sufficient.

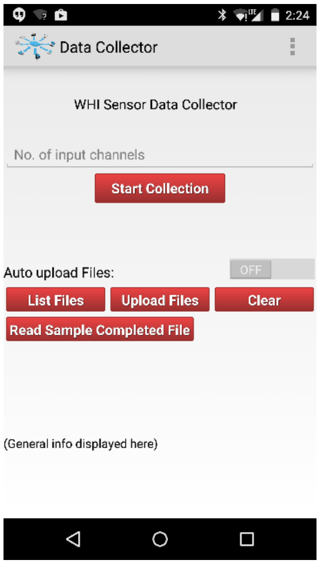

Figure 21: Screenshot of Android Data Collection App

Here we take the time to elaborate more on the implementation details of the automatic and manual uploads. Currently, the automatic upload will delete the files from the phone immediately after a successful upload. This is because the data will start to take up significant space, especially when sampling at 8.192 kHz. In the manual upload, we assume the user has a good idea of the data storage resources that they plan to use, so we defer these considerations to their best judgement.

Both auto and manual upload schemes use the same parsing and file-writing schemes. First, the bytes are parsed in real-time as the data is coming in. There are two key bytes schemes that the app is looking for and both are composed of two bytes. These two schemes are the byte headers that were discussed in the previous section: the RTC timestamp header and the data header. As bytes first come in for parsing, all of them are discarded until a RTC timestamp header has been read. This is done so that the start of the official data begins at the start of a second. The bytes that are discarded, at most would only be a second's worth of data, and are expendable. If a data header is read, the number of bytes read next is two times the number of input channels selected. Assuming no dropped data, the data header reading process is repeated until a full second of data has been completed. At this point, we would expect a new data timestamp, however, the app only reacts to the bytes read and parses them based on the headers. If we discover post-processing that the byte order has deviated from the expected scheme, then we can conclude that there were dropped packets at some point in time. All that needs to be done to determine if the data is complete is to count all the samples per second that arrive and identify that the timestamp header was read.

As the data is parsed, it is written to a file on the smartphone in a .csv format. The name of the file contains the time at which it was started in "YYYYMMDDHHMMSSsss" form which is resolved down to the millisecond. The first column in the file represents the RTC counter timestamp. However, a number timestamp only exists in that column for a row that describes the first sample of a second of data. The data for each channel is separated by a comma and each sample for a set of channels is separated by a newline character. The 2-byte data is converted into a decimal value between 0 and 4095, inclusive, which reflects the 12-bit ADC range. This format was chosen because many other data processing platforms are compatible with this format such as MATLAB and Microsoft Excel. Additionally, it is easy to read the data if one wishes to glance at it in a text editor.

```
6,0,0,0,775
,0,0,0,775
,0,0,0,783
,0,0,0,783
,0,0,0,783
,0,0,0,759
,0,0,0,759
```

Figure 22: Example Data File with First Three Input Channels Shorted to Ground and Last Channel Floating

4.2 Pushing the data to the server enterprise for post-processing.

As the data is collected, converted and written to the files on the smartphone, the next step is to push the data to the server enterprise. The app uses the wireless method that is available to it, whether it be Wi-Fi or cellular service. For each file, we use a HTTP put command and a PHP script at a designated location on the server which receives the data in the file format explained above, and saves it to the hard drive of the server. One thing to note is that the file formatting can be done on either the smartphone or the server, as both have the capability or potential to do so, it is just we don't want the embedded node to do this work as bandwidth, battery life, and memory are much more limited compared to the other nodes. After data transfer between the smartphone and server is complete, multiple actions can be taken. The first is to download the data onto a personal computer for viewing and analysis in a program like MATLAB. Another other option, at least for product and service needs, could be to run a program on the server that uses an algorithm to analyze and or calculate certain metrics of the collected data. While the platform and architecture for the end-to-end data collection system has been implemented and detailed, these next-step signal-processing endeavors are not covered in the scope of this paper.

Chapter 5. System Results

This section shows and discusses the results of the *AbStats* platform. Figure 23 shows a sample of a abdominal signal due to peristalsis that is recorded using *AbStats*. Some characteristic features of this signal include large signal energy compared to the other temporal segments, as well as lots of signal power, because the signal energy is captured within a relatively short time, on the order of milliseconds. It is these signals that are trying to be identified and accurately counted to determine the patient's current digestive health.

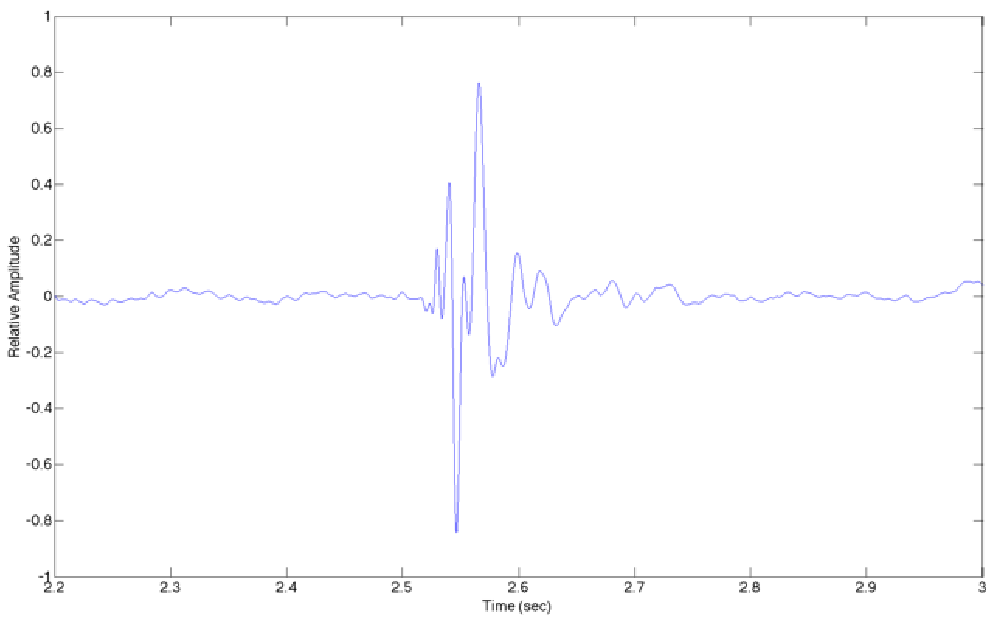

Figure 23: Healthy Subject Signal Collected From *AbStats* [15]

Figure 24 shows a 600 second recording using *AbStats*. One can observe that there is lots of signal activity. The signals appear to be very sporadic, but very easy to identify qualitatively. Although the signals aren't purely periodic, one can calculate the rate at which the peristalsis event signals are occurring to help diagnose digestive health.

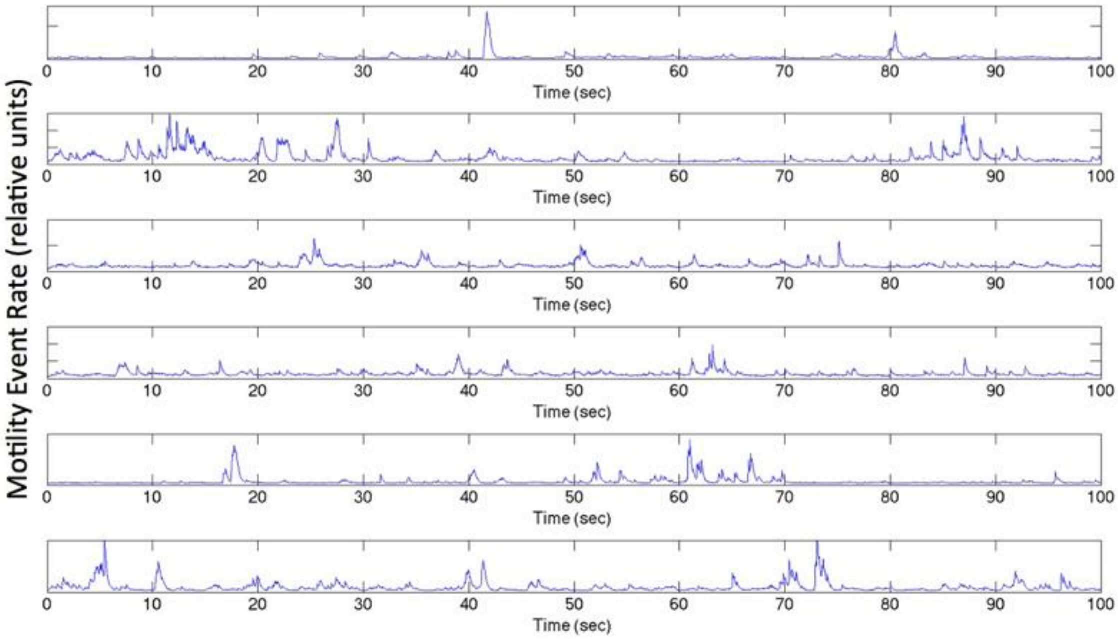

Figure 24: 600 Second Event Record for Healthy Subject

Figure 25 is a 600 second recording for a post-operative ileus subject. Overall the acoustics recorded from *AbStats* show a very quiet signal. There seems to be some peristalsis signal activity, but it is extremely rare and surrounded by periods of non-activity. The very clear differences between the data of a healthy and post-operative ileus subject point to a promising, reliably way of distinguishing between the two types of patients.

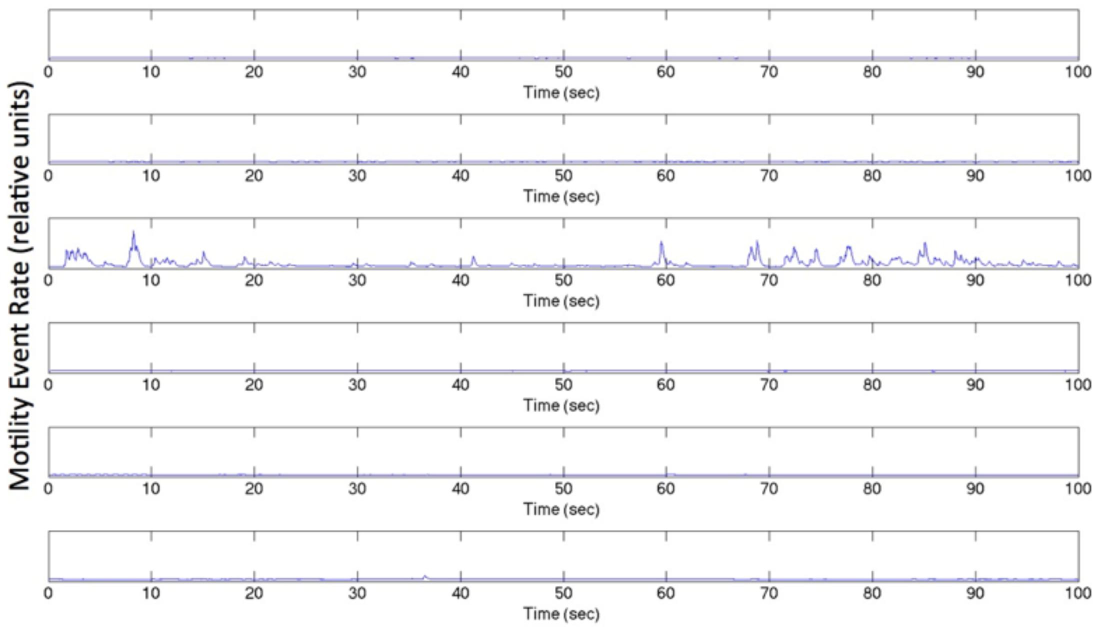

Figure 25: 600 Second Event Record for Post-Operative Ileus Subject

Lastly, Figure 26 shows the motility rate for the different human conditions. The results are astonishing. *AbStats* is able to clearly distinguish between three types of statuses during the post-surgery recovery process. That is, the overlap between the rates of these three status are separated very clearly, and can give a confident evaluation of the patient's digestive health by monitoring the motility rate determined from *AbStats*. It would seem logical that if the patient is following the proper nutrition intake method, they should have an optimal progression from the post-surgery, no nutrition state to the healthy subject state. The benefits of the solution to the prolonged length of state problem have completely improved how post-operative healthcare is delivered. Prior to *AbStats*, physicians weren't even sure if they were on the right track to getting a subject to a healthy status. Now, with the application of *AbStats* there may even be potential to optimize the recovery process, saving millions of dollars in the process.

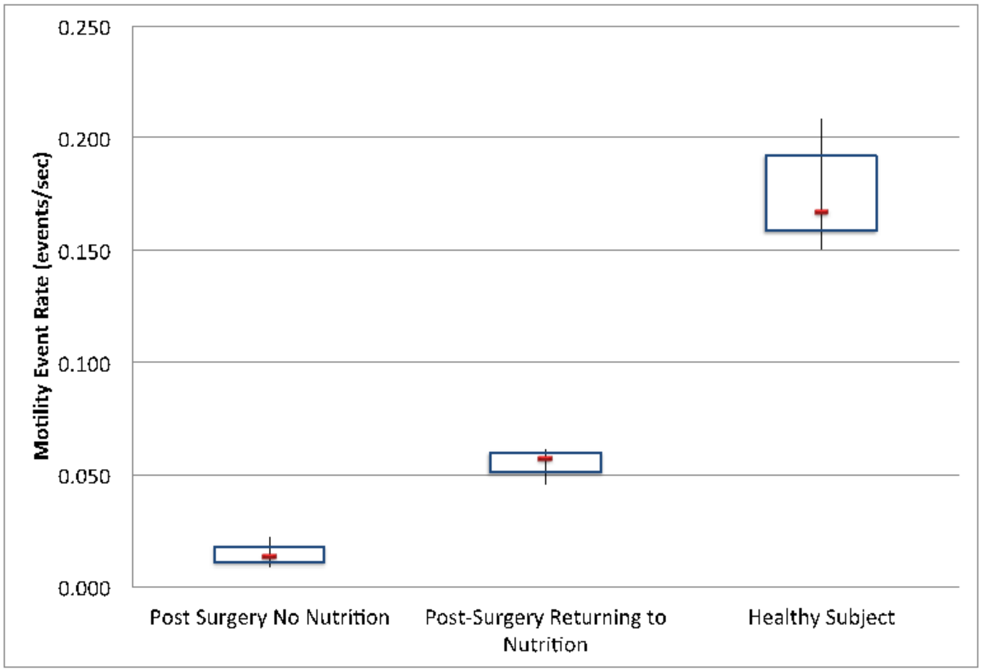

Figure 26: Motility Event Rate for Different Human Conditions

Chapter 6. Future Work: Effects on the wireless health market and expanding application opportunities.

6.1 Veterinary and Equine applications

Wireless AbStats has great potential for equine applications. The problem with horses, particularly race-horses, is that they may develop a form of ileus in their digestive tract. However, it takes a skilled veterinarian to detect the disease by listening to it via stethoscope. The biggest problem is that by the time (if the veterinarian even makes it in time) the veterinarian even detects the disease, it may be too late for the horse. In most cases, they have to fly the horse out immediately to a special facility and conduct surgery. While these costs are enormous, the autonomous nature of *AbStats* can be deployed and detect these diseases with much more vigilance. While the system capabilities are present with the advancements to *AbStats* and *Wireless AbStats*, the missing part of the equation are the signal processing algorithms, specific to the species being study. In the case of the horse, an understanding of the digestive tract is required, as it is much longer than that of a human and the temporal occurrences of peristalsis events are much different than that of a human.

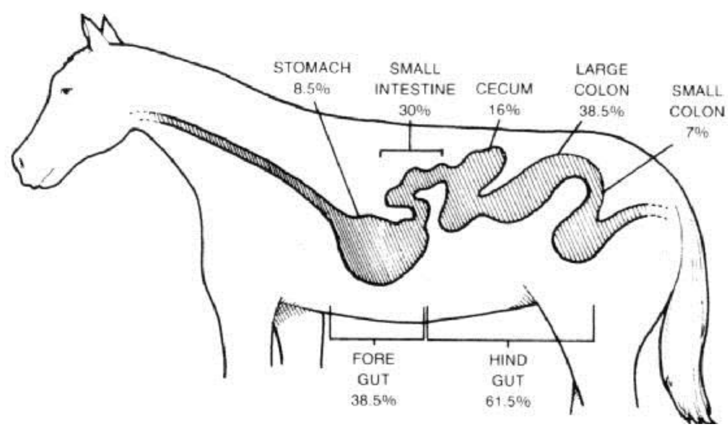

Figure 27: The Equine Digestive Tract

Despite these challenges and differences, there is still a great opportunity to develop these algorithms, as no such attempt at recording and digitizing the audio sounds of such animals, especially in equine applications, has ever been done before.

6.2 Neonatal intensive care unit applications

Just recently, there has been an initiative to cater the *AbStats* system for neonatal applications, specifically for use in the Neonatal Intensive Care Unit (NICU). A typical problem in neonatal care is that the infants need to double their body weight by a certain deadline. However, the only form of nutrition that is compatible with the infant is mother's milk. If an intravenous therapy is administered, the infant is likely to die. Thus, proceeding with the mother's milk solution, they deliver the milk directly to the infant's stomach via a tube that runs through their mouth and down into their stomach. However, since they don't know how much milk to deliver, they wait a while and re-apply the tube and try to detect how much milk was left to give more accurate feeding sessions. In short, the process is very risky, because if the infants do not meet the weight gain deadline, they are sure to die. The physicians involved with the NICU are confident that *AbStats* can provide a useful alternative to the detection of the infant's digestive operation.

Figure 28: Infant in a Modern Neonatal Intensive Care Unit

Next is a description of a prototype for the NICU application. Here, the focus is not so much on the *AbStats* system, but developing a mechanical interface that connects to *AbStats* that meets the strict requirements unique to the NICU setting. Here, we drilled an array of holes in the top of an ABS box. The diameter of the holes were made slightly larger than the inner diameter of the sensor housing. This allows the sensors to be suspended face-up into the ABS box. Cables run beneath the sensors, inside the ABS box and are connected to 3.5mm female jacks that are installed in the side of the box. The reason for this base setup will be apparent shortly. Next, gaskets are placed on top of the sensors. The gaskets are made out of silicone of the lowest durometer. This ensures compliance with whatever material is placed above it. On top of the gasket layer, a 12 in. by 7 in., extremely-low durometer silicone gel cushion is placed. This layer serves two purposes: the first allows acoustic signals to propagate through to the sensors; the second is that it serves as a cushion for the infant to lie comfortably on. This ABS box and silicone gel cushion setup is enclosed in a sterile sleeve that is commonly used in the NICU.

Figure 29: Acoustic Platform to Measure Data Safely From Infants

Below, in Figure 30 below, is a sample signal recording of a person using both the acoustic platform rig in Figure 29 and the classic *AbStats* system at the same time. Figure 30 demonstrates the practicality of using such an acoustic platform rig setup and clearly shows how the acoustic signal is able to be coupled from the human body, through the silicone gel, and to the electret mic sensors.

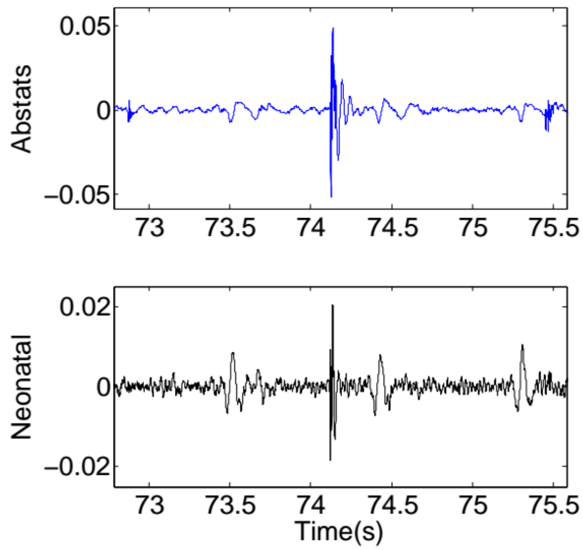

Figure 30: Simultaneous Signal Recording for Classic *AbStats* and Neonatal *AbStats* Rig, Top – Classic *AbStats*, Bottom – Neonatal *AbStats* Rig

6.3 Athletic applications and optimal nutrition

Extremely competitive athletes are under a strict diet regimen in order to improve performance. Depending on the sport there may be certain considerations on what types of food to eat, when or how often, and of how much. It may be possible that the form factor of *Wireless AbStats* may facilitate the discovery of insights that can lead to athletic competitive advantage. Additionally, the acoustic technologies presented by *AbStats* can also collect data on cardio and respiratory sounds. While cardio and respiratory sounds are out of the scope of this paper, the main take-away is that the wireless architecture of *Wireless AbStats* presented in this paper can make athletic monitoring possible.

References

[1] T. Asgeirsson, et al. *Postoperative Ileus: It Costs More Than You Expect*. Journal of the American College of Surgeons, Vol. 210, No. 2 (February 2010), pp. 228-231.

[2] A. J. Senagore. *Pathogenesis and clinical and economic consequences of postoperative ileus.* American Journal of Health-System Pharmacy, Vol. 64 (October 2007), pp. 53-57.

[3] E. H. Livingston and E. P. Passaro Jr. *Postoperative Ileus*. Digestive Diseases and Sciences, Vol. 35, No. 1 (January 1990), pp. 121-132.

[4] *Bleeding in the Digestive Tract*. National Digestive Diseases Information Clearinghouse, National Institute of Diabetes and Digestive and Kidney Diseases, National Institutes of Health, Bethesda, MD (2013).

Available: http://digestive.niddk.nih.gov/ddiseases/pubs/bleeding/.

[5] H. J. Ehrlein and M. Schemann. *Gastrointestinal Motility*. Technical University of Munich, Munich, Germany.

[6] R. Ranta, et al. *A Complete Toolbox for Abdominal Sounds Signal Processing and Analysis*. 3rd European Medical and Biological Engineering Conference IFMBEEMBEC (2005).

[7] J. D. Hardcastle and C. V. Mann. *Study of large bowel peristalsis*. Gut (1968).

[8] *Welch Allyn Harvey DLX Double Head Stethoscope Navy*. Steels, Saint Joseph, MI (2014). Available: http://www.steeles.com/products/welch-allyn-harvey-double-headstethoscope-

5079-327.

[9] *Electret Condenser Microphone: XCM6035 series* Datasheet. SPL, Hong Kong (2008).

[10] *Amplified-Mic-Electret-v14.* SparkFun, (May 2010).

[11] *OPA344 Low Power, Single-Supply, Rail-to-Rail Operational Amplifiers.* Texas Instruments, Dallas, TX (April 2000).

[12] *MSP430x5xx and MSP430x6xx Family User's Guide*. Rev. L. Texas Instruments, Dallas, TX (June 2008).

[13] *DS3231 Extremely Accurate I2C-Integrated RTC/TCXO/Crystal.* Rev. 9. Maxim Integrated, San Jose CA (2013).

[14] *Bluetooth Data Module Command Reference & Advanced Information User's Guide.* Version 1.0r. Roving Networks, (March 2013).

[15] B. Spiegel, et al. *Validation of an Acoustic Gastrointestinal Surveillance Biosensor for Postoperative Ileus*. Los Angeles, CA (2013).