Abstract—In this paper, the hardware needed to run this system is first explained. An electronic stethoscope will be created and designed. This will allow the analog signals of a heartbeat to be digitized for robotic, and networking uses. The stethoscope will be designed for both easy robotic, and human use. The robot that will be used in this research is Baxter from Rethink Robotics. Baxter, as well as many other robots arms, or humanoid robots has and end gripper or hands. Baxter hands works very similar to most other ones. So, the device created should also be able to be used by other generic robot arms, or humanoid robots. Then the second half of this section is used to explain the software implementation needed to run this system. The software portion of this system will be implementing database design, hardware coding, and network coding. The database will use the standard Structured Query Language (SQL) with the specialized subsystem called PostgreSQL. The hardware coding will be done using C++ on the Particle Photon micro-controller. The network coding will be using one of the industry standards web programming languages JavaScript. More specifically node.js, which will be used for transferring data from one source to another over WiFi. This will allow a web interface to display a patients heart data on a web-page.

I. INTRODUCTION

Taking a persons vitals during a doctors visit is the corner stone of medical appointments, which always checks for body temperature, their pulse rate, respiration rate (breathing rate), and blood pressure. Hospitals see hundreds of people a day, and humans can only act so fast when taking vitals given the amount of staff working. The development of a method to record human vitals (mainly the human heartbeat) with the help of robotics, and the internet of things (IoT) could help solve this problem. The design objective of this research is to allow a humanoid robot to take the vitals of a patient/volunteer instead of a doctor, or nurse. Since humanoid robots are quite useful for many general tasks. If the vital recording tool is a good enough fit the robot should be able to use this tool just like a human would. However, in some circumstances for this to be possible some custom hardware, and network coding must be implemented.

II. RELATED WORKS

Robots are starting to become more frequent as technology evolves. With the overall price drop that comes from that. Multiple companies, and universities have been trying to create robots to solve specific problems. Since the medical industry is one of the largest industries out there. There have been quite a few attempts to create robots to solve these in a hospital.

Most of these problems are caused by humans having to preform menial tasks that divert their attention from the patients. The company Diligent Robotics is trying to alleviate this with the help of mobile robots. Over the past year, Diligent Robotics has been testing their robot named Poli in several hospitals in Austin, Texas, where its learning how to help nurses with simple fetching tasks” [1]. Since hospitals are pretty active with patients this could help both the doctors, and nurses save time which could be used seeing more patients.

Another task that is being tackled with the help robots is drawing blood. The company Veebot has created a robot that accurately takes a patients blood [5]. It can do this very accurately, and quickly using image processing. The process starts when the patient inserts his/her arm into a sleeve like hole. After that it restricts blood flow to make the veins easier to see. This is a pretty great solution in a hospital since blood is taken hundreds of times a day. Also if it can find a vein then it could also work for IV’s. Which can be considered another menial task.

Another problem is stress, and it can take place in and out of hospitals. The company behind PARO have created an animatronic robot to help combat this. The robot is modeled after a baby harp seal [4]. The robot is designed with an AI to move, and make sounds when interacted with. The AI tends to learn what personality the user likes the best, and adapts to that. Many users just can’t seem to resist this robots charm. Which having a better way to calm down patients besides sedation is a amazing humane advancement.

III. PROBLEM STATEMENT

The overall goal of this project is to be able to take a humans heartbeat. Each single person that goes to a hospital usually has to have this done. This can take a good amount of time too, depending on the situation of the facility. So the idea is that this project could help take some of the weight off hospitals by eliminating this menial task.

Hundreds of people go the doctor a day, and mostly all them need a pre-screening before they can see their physician. The pre-screening process involves taking multiple types of human vitals. These types consist of the heartbeat, blood pressure, temperature... etc. If a hospital is under staffed then this process can take much longer than expected. Most of this time the patient will spend waiting. Which is time that could be better used for both the patient and the hospital staff. Another problem is how all of this data is
shared. While yes most of this goes into a database. The process involved still usually involves a human. This is a menial that could be done by implementing an IoT network to do this for them.

The heartbeat produces a sound, and this sound can be broken up into multiple specific frequencies. Which to find these frequencies a Fourier transform can be applied [3]. Once this has been found a system can be designed to zero in on these frequencies. This system takes in that analog signal, isolates the wanted frequencies, and then digitizes for later usage. With the heartbeat signal in a digital format it can now be sent over the web. Which allows it be stored for later use, displayed on screen for on demand viewing, and sent to a robot.

With this process now being automated. Thus eliminating a couple menial tasks, for example entering data in to the database, and the simple pre-screening process. The staff can have more time to interact with the patients. And now with an IoT network the entire hospital would be more connected, and have easier viewing of a specific patients vitals.

IV. METHODOLOGY

The structure of this project as a whole is shown below in the form of a flow chart. This chart can be broken up in to a couple of main features. Those features are the heartbeat sensor, the website, and robotics implementation.

V. HEARTBEAT SENSOR

In order to retrieve heartbeat data that can be sent over the web a custom sensor was designed. The sensor uses a microphone pre-amplifier, bandpass filter, and WiFi compatible micro-controller

A. Microphone Pre Amplifier

The human heartbeat is an analog signal, which we need to digitize with a transducer. The choice to use a microphone as the transducer is heavily based on the principle of how stethoscopes works. A stethoscope is a very sensitive pressure sensor made of of several components. The most important being the bell diaphragm. Which when pressure is applied air is transferred through the tube to the ear pieces [2]. Since the bell is outputting air, this allows the signal to be picked up by a microphone.

The chosen type of microphone was an electret microphone. This was because in order to accurately "hear" the heart. This microphone needed to be inserted into the stethoscope tube at the closest possible position next to the exit hole of the bell diaphragm, as well as be very sensitive since the air coming out of the bell would be very minimal. Electret microphones fit both of these criteria, and the one used has a diameter of 6mm, and sensitivity of -44dB

The signal output from this microphone is very small, and needs to be amplified. The amplifier designed here is calculated to be 68db. Since Electret microphones have an internal JFET the circuit design must consider how to properly bias this JFET, because with out this the signal output is null.

Choosing the components for the above circuit is the most important part, and needs to be calculated for the exact microphone specs.

<table>
<thead>
<tr>
<th>TABLE I</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELECTRET MIC SPECS</td>
</tr>
<tr>
<td>Sensitivity</td>
</tr>
<tr>
<td>Vs</td>
</tr>
<tr>
<td>Imax</td>
</tr>
<tr>
<td>Z(Ω)</td>
</tr>
<tr>
<td>S/N</td>
</tr>
</tbody>
</table>

To start this calculation find out how many millivolts will be generated per pascal of pressure.

\[
\frac{mV}{Pa} = \frac{10^{sensdB}}{20}
\]

(1)

next find the current generated by the pressure

\[
\frac{uA}{Pa} = \frac{mV/pA}{Z}
\]

(2)

next find the max current that is generated by the pressure

\[
uA = \frac{uA}{Pa} \times 2
\]

(3)
The value of resistor R2 can be found using the max current. We use 1.228V for 100dB or the maximum pressure expected.

\[ R2 = \frac{1.228V}{U_{a_{\text{max}}}} \]  

(4)

The value of C2 can now be found. Assuming f as 133KHz as the desired pole for the filter. This is a feedback resistor, that will keep the system stable from the parasitic capacitance of the opamp.

\[ C2 = \frac{1}{2\pi f R2} \]  

(5)

R1 can be found by using node analysis. Where Vmic is the required voltage the JFET needs to be biased, and Is is the needed current.

\[ R1 = \frac{Vcc - V_{\text{mic}}}{I_s} \]  

(6)

C3 can be found using the same method to find C2, but substituting R2 for R1, and f for the chosen corner frequency

\[ C3 = \frac{1}{2\pi f R1} \]  

(7)

Applying the same steps to the other components will yield similar results, and help choose the correct values. Once this has all been done, the circuit should have a properly biased Electret microphone feeding in a signal to an opamp. Which is then amplified to a level which can be used. However since this microphone is picking up all the noise, and the amplifying it. The resultant output is quite noisy, so filtering must be done.

B. Band Pass Filter

The filter design is a 2nd order butterworth with a quality factor of .85. The specific type is a multiple feedback low pass, in cascade with a multiple feedback high pass. This done to create a nice flat plateau in the frequency response from 100hz to 1KHz.

\[ BPM = \frac{\Delta Max \times \text{SampleRate}}{60 \text{seconds}} \]  

(8)

D. Particle Photon Code

First the sensor has to determine if the input signal has passed a threshold. Which indicates that the sensor has been placed in the correct position. This is done in conjunction with the robotics system. To start the sensor will take a quick 1 second heart reading, and store that in to a temp array. Once that array is full it will iterate through it until a global maximum is found. This is then stored to a variable, turned into a JSON string and sent to the robot over UDP/TCP to determine if the threshold has been reached.

If the threshold has been reached a full 3-5 second reading is taken, and stored in to the main array. The array is iterated through to find the global maximum, which then is used to normalize the array. Once this has been found a tolerance level/range will be created. The array will then once again be iterated through, but this time starting at the global maximum. This continues until a value in the tolerance range is found. Which then it uses both points to take the difference in time, and then divide by a minute to find the BPM.

\[ \text{BPM} = \frac{\Delta \text{Max} \times \text{SampleRate}}{60 \text{seconds}} \]  

(8)

If this BPM is in a reasonable range then the heart data array is turned in to a binary string, and added into a JSON String. This JSON string also includes the calculated BPM. Then using UDP/TCP this String is sent to the IP of the raspberry pi for it to added in to the postgres database.

VI. WEB PROGRAMMING: BACK END

A. PostgreSQL

SQL is the main language for databases. Whether that be inserting, selecting, or even deleting records. There are many different types of frameworks that help implement SQL databases. The one this project uses is called postgresSQL. Postgres is a object relational database management system (OBDBMS) and its main focus is helping users create database servers. These database servers are quite nice for web interfacing, as it can handle many concurrent users at once. As well connect to PostgREST Which is a REST API that can be generated from a postgres database schema. This is really good for prototyping and designing a structure for web queries.
B. NodeJS

Node Js is a runtime framework of JavaScript. This specific framework is used to run JavaScript code on a server. This framework is becoming one of the most used in web programming, being used by Microsoft, PayPal, etc. This framework allows the creation of webservers, which are known as nodes. These nodes are then allowed to run the core function of the network. Whether that be file reading, I/O over the network, or just database queries.

C. ExpressJS

However NodeJS can only run on a server, thus making it difficult for use via client side i.e., a webpage. So another web framework for nodeJS was created with the name of ExpressJS to solve just this. ExpressJS is a NodeJS Web application framework. It works by creating server side routes which are URL handling codes. These codes are then declared as a function in another file JS file. So when these codes are executed client side it can access the server side NodeJS scripts.

VII. WEB PROGRAMMING: FRONT END

A. ReactJS

ReactJS or React is another JavaScript library. This library is used for creating user interfaces over the web. The framework is component-based. Meaning that changing the elements states of the webpage can be done dynamically, and quickly. The scripts can also be written in a Document Object Model (DOM) similar to how HTML and XML work. Which is great for displaying and changing database records from a webpage.

B. XMLHttpRequests

In order to have access to the server side code, specifically being the database, XMLHttpRequests need to be implemented. An XMLHttpRequest (XHR) is part of the JavaScript API which allows programmers to send data back and forth between the front and back end of the webserver. Similar to how curl works in CLI.

VIII. ROBOTICS

A. ROS

ROS stands for Robot operating system and is used in robots around the world. ROS can only be used on Linux platforms, and can be used to simulate, or control a real robot. ROS is built on the foundation of networking. Which works off the premise of publishers and subscribers.

Publishers are a way to send out data to the public. The public in this case is a ROS topic. A ROS topic, can be seen as a specific topic that is set up in the code, or in a real sense a post office. The next portion is the subscriber, a subscriber can be seen as the townspeople. The townspeople will get a message that their mail is there, and then will go to the post office. In short ROS works off messages being posted to a topic by publishers. Then the subscribers who are looking for the messages on that topic will be pinged that there is a new message on that topic. Then the subscriber will take that message and run its callback function with that data/message.

B. OpenCV

OpenCV is currently the best library out there for computer vision. OpenCV is an open source library for implementing computer vision. Computer vision is the method for helping a computer "see". The computers are allowed to "see" using cameras, and taking the input from these cameras and running functions on them. Allows us to tell the computer/robots what is around them, and what they should do in that environment. OpenCV has many functions to help with this, including face detection, finding contours, and even face recognition. It is also fully compatible with ROS.

C. Point Cloud

A point cloud or (PCL) is a method for viewing a set of 3D data points. These data points are usually found/taken from a 3D camera. While openCV is great at finding requested objects, it can only do this in (X,Y) cords. However, if the openCV data was used in conjunction with a PCL the full (X,Y,Z) cords of a desired object can be found. For this project this was implemented to find the position of the human heart.

IX. RESULTS

A. Results: Heartbeat Sensor

The pre-amplifier circuit works very nicely, with the values of components that were chosen. The JFET is properly biased, thus allowing output from the microphone. This output is quite small around 3-5mV, and is then amplified to a maximum of 10V.

![Frequency Response of pre-amplifier](image)

The multiple band pass filter that was created by cascading a LPF, and HPF. The first design used a .7 quality factor, however it didn’t have a steep enough falling off rate. Thus it was changed to .85. This design is implemented using the filter type of butterworth, and an order of 2.

The chosen center frequencies are based on the sounds of the heart. Taking an FFT of these signals allows us accurately design our filter to fit this range[3]. There are 4 main sounds of the heart S1, S2, S3, S4. When both sides of the hearts muscles open and close these sounds are made. S1 as we can see is the loudest, and takes place when the muscles close on the right side of the heart. This part of the heartbeat, much like the others has multiple frequencies to design for.
Instead of creating many BPF for each one. A single wide bandwidth BPF was created.

After the signal is filtered which also lowers the voltage to around 3-4V it needs to be regulated. This is to avoid accidentally blowing up a micro controller analog input. The regulation is done with a 3V zener diode in parallel with a 1KΩ resistor. The final step is compensate for how the analog to digital converter(ADC) in the particle photon works. These inputs do have a internal ADC however, the inputs do not work with negative voltage values. In order to alleviate this the circuit below is implemented.

The particle photon code was implemented, and was able to find a BPM using the method discussed earlier. Using Putty, there serial monitor was able to print out the sensor data. Which is plotted below. The graph is more visual way to view the array’s elements.

Our NodeJS scripts for accessing the database is stored in queries.js. Where all of the declared routes are defined here. This will start by creating a connection string to connect to the database, and query using that connection as well. For security reasons the connection string allows us to "login" with a existing account.

C. Results: Robotics

Our robot of choice Baxter runs by using ROS. Baxter needs to be able to find the patient in front of him, and then that persons heart position in (X,Y,Z) cords. By Using a face detection node, and then using the radius of the face detected. A ratio of how far the heart is from the face origin can be found. This is done using OpenCV libraries, and then the cords are published to a rostopic "heartxy"

Now that the (X,Y) cords have been found they can be used to find the Z cord. Since the camera Baxter is using to see is a kinect this allows the use of 3D camera data, and the point cloud library. The (X,Y) cords are found by subscribing to "heartxy". Then they are sent into a function that takes in X,Y a point cloud and returns a Z cord. This is then sent out on a new rostopic "heartxyz"

Baxter’s main computer is subscribed to this topic, and uses these cords in conjunction with a python library of forward kinematics to place the sensor on the patient. Once this sensor is placed a message is sent to our sensor over UDP/TCP. This message tells the sensor to run the threshold test.

The threshold test will take a short reading to find a maximum value. This Max value is then sent back to our robot. Where Baxter will decide if the reading has reached a
threshold. If so Baxter sends another message to our sensor to indicate that the full reading can be taken. However, if the reading doesn’t reach the threshold. Baxter will move the sensor up or down and then repeat the same test. This is done until a suitable reading is found. Baxter also uses previous test iterations to tune in to the correct direction of the heart.

X. Future Work

In the future improving the bandpass filter to filter for a smaller bandwidth would help eliminate the unwanted spikes from the sensor output. This system is very variable/expandable since a humanoid robot is the robot of choice. Meaning if more sensors were created to find other vitals such as blood pressure. The pre-screening process could be shortened even further.

XI. Conclusion

The individual components, and the choice in technology worked well. Same can be said for having all the individual components talk to one another. Creating a sensor to talk to robots, confirms the idea that a robotic doctor/nurse could be a possible future piece of technology. The results are promising as well, since both humans and robots can use the custom sensor to take their vitals. The use of IoT, and robotics allowed the system to work on it’s own, and on the demand of humans the data could be shown on any device.

REFERENCES