Current Work:

This week we began to program and display the results from the pulse oximeter and thermometer probes. In terms of displaying body temperature, the thermometer we purchased takes about 30 seconds for the voltage level to stabilize. Since our C compiler only allows a delay function with a maximum of 200 ms, we had to find a reasonable way to sum these functions up to reach the 30 seconds required. This was accomplished with a simple delay function embedded into a while loop. Since the Vin voltage value is required for the equation, we used the result of the A/D converter into the following equation:

$$5.0 \times \frac{A/D \ Result}{1024}$$

Where 5 is the reference voltage and 1024 for a 10 bit A/D converter ($2^{10}$). The result of the A/D converter is then passed into the temperature equation which Jenna developed to give the corresponding result. Initially, we were obtaining wacky results when passing the A/D result into our temperature equation. We later realized that using the integer 5 instead of 5.0 was something that the compiler did not like. Below is a picture of our working thermometer.
We also were able to get the pulse oximeter value to display on our LCD. Due to calibration issues, we were not able to accurately read a patient’s SP02 level, but rather infer from the voltage output when a person’s SP02 level reached 98%. A few weeks ago, Jenna came to the conclusion that at voltage levels above 250mV, the pulse oximeter has read 98% blood saturation. So, in our code we waited until the ADC value reached the appropriate integer of $51 \left( \frac{250}{5} \right) \times 1024$ using a simple while loop:

```c
while(n==0)
{
    ADON=1;
    ADGO=1;
    while(ADGO)continue;
    ADIF=0;
    adcvalue=(ADRESH<< 8) + ADRESL;
    DelayUs(50);
    if(adcvalue >= 51)
    {
        LCDwritestring("SP02 Level: 98%");
        n=1;
    }
}
```

To program the respiratory rate and pulse, we need to utilize some sort of clock function or timer in our compiler. Last night was spent trying to program a clock function in C which was quite unsuccessful. I do have some ideas which I am going to try this week which will hopefully work out. I also began working on the PCB and schematic for our design. Hopefully by the end of the week we will have all our hardware finalized so we will be able to order it and have it in by Tuesday.

Future Work:

This coming week we plan to finish displaying the vitals signs on the LCD and move on to programming the Bluetooth module. Since no one has ever worked with a wireless system before, this could turn out to be quite the challenge. Since we already have the code for RS 232 communication, all we have to do is figure out how to send commands out to the module. Code for the speech module is completed, except it is not quite working yet. I predict this is only a minor error, and debugging that problem will not take too much time.
Project Review:

These past two weeks I have learned and accomplished a significant amount in terms of microchip programming. I hope to have the rest of the probes done by the end of the week, leaving just the Bluetooth programming and the packaging of the device. Mike has been doing research on enclosures as well as the buttons we need. He also has been working on the RERC website required for the competition. Jenna finished the probes last week and has been assisting me with the programming. With hard work, I expect this to be completed by the deadline of April 27th.

Total Hours Worked: 29