Peter Campellone
May, 10 2015
ELE205/6 Final Project Report

Handheld Medical Diagnostic Tool

Abstract

The goal for my final project was to create a handheld device capable of taking multiple medical measurements and displaying them to the user in a timely fashion. Furthermore, this device was designed with the goal of being able to last a full work day of approximately 8-10 hours on a single charge. The measurements this device would be able to take are heart rate via a plugged in heart rate sensor and temperature via an IR sensor on the left face of the device. These measurements would then be displayed to the user via a 16x2 LCD which would be programmed with a menu system consisting of various options. The menu would be controlled via two tactile switches placed to the right of the LCD screen. All of these components would be housed inside a custom designed 3D printed case.

Methods

The first task for creating the handheld medical diagnostic tool (HMDT) was to create a bill-of-materials (BOM) of all the components and parts needed to create the device. Below in Figure 1 is the original BOM for this project.

Part	Quantity	Price
Atmega 328p Chip	1	$1.63
16x2 LCD	1	$5.98
Pulse Sensor	1	$34.68
Temperature Sensor	1	$6.13
2000mAh Li-Po Battery	1	$24.61
5V Step up Board	1	$8.90
Micro USB Li-Po Charger	1	$4.57
3D Printer Filament		$28/kg
Assorted Misc. Electronics	1	$15.00

Figure 1. Original Bill-of-Materials

From the BOM the minimum total estimated cost of the HMDT to build would be $115.50, when figuring a usage of about 500g of 3D printer filament totaling $14.00 for the filament.

After creating the BOM and ordering the necessary parts the next step was to create the code that would run this project. All of the coding was done using the Arduino IDE exclusively, which helped keep everything simple. To run the LCD, Temperature Sensor, and Pulse Sensor it was decided that the best and easiest way was to use the open-source libraries that had been already created for these devices. For example, both the LCD and temperature sensor use the I^2C communication protocol therefore instead of creating a new library and new code to make these devices work it was simplest to use the available open-source libraries. The library for the LCD, called “New Liquid Crystal” was provided by F. Malpartida and the library for the temp. sensor was provided by Adafruit Inc. Furthermore, the library and code for the pulse sensor was provided by World Famous Electronics LLC. (Credit and appropriate links is in final source code). Using these open-source libraries just made it much easier to get these devices up and running because nothing beats having openly available code that is known to work and work well at that. The rest of the code for the menu system, button reading, I/O, and other menu options is entirely unique and was created specifically for this project. Please find the final and fully documented source code included in the .ZIP file.

The next part of this project was to test the code and all of these devices together. The initial test would be done while running the entire project connected to the Arduino UNO. The second test would be done with the final board featuring the lone Atmega 328p Chip on a protoboard and all the connections to the other components. This second test is discussed later on in the report. Initial testing proved that the code I had created did in fact work besides for some minor display errors. In the end the code only needed minor tweaking in the form of modifying LCD output commands, introducing or changing delays, and optimizations. In the end the final code had four distinct modes as described in Figure 2. below.

Mode	**Function**	**Other Functions:**
1. All Off	Idle State, No Measurements	Turns off LCD backlight after 25 seconds (configurable)
2. Temp. Sensor	Turns on Temp. Sensor, Reads output from Temp. Sensor	Displays temperature value to LCD in Fahrenheit (configurable)
3. Pulse Sensor	Turns on Pulse Sensor, Reads output from Pulse Sensor	Displays heart rate to LCD in BPM
4. Battery Charge	Displays the current battery charge percentage.	Displays a message if the battery goes below 20%

Figure 2. HMBT Table of Modes and their Functions

After completing the code and testing the device with the Arduino Uno and a breadboard the next step was to design a case for the device. The case was modeled using a 3D CAD program called Tinkercad which is a simplified and free online version of Autodesk 123D Design. In the end six versions of both the top and bottom cases had been created for the HMBT. Overall, the first three iterations were for determining the general shape and overall design of the case while the last three were based on test fitting the components within the case and making changes. Below in Figure 3. are some of the case designs that were printed. In the .ZIP folder you will find the files for the final designs of the top and bottom case. These files can viewed at www.viewstl.com.

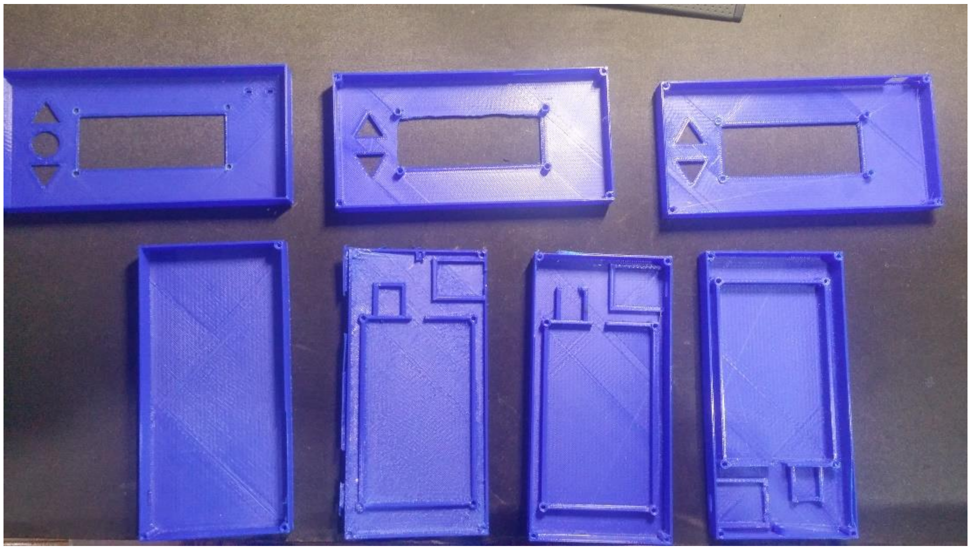

Figure 3. Some of the test cases that were printed

The last thing to do to complete the HMBT was to assemble the device and test it for the second time in a somewhat assembled state. Figure 4. and Figure 5. show the device during assembly. While Figure 6. shows the device during it’s second test phase. Also, in the .ZIP folder there is a video showing the device functioning and being tested before being fully assembled.

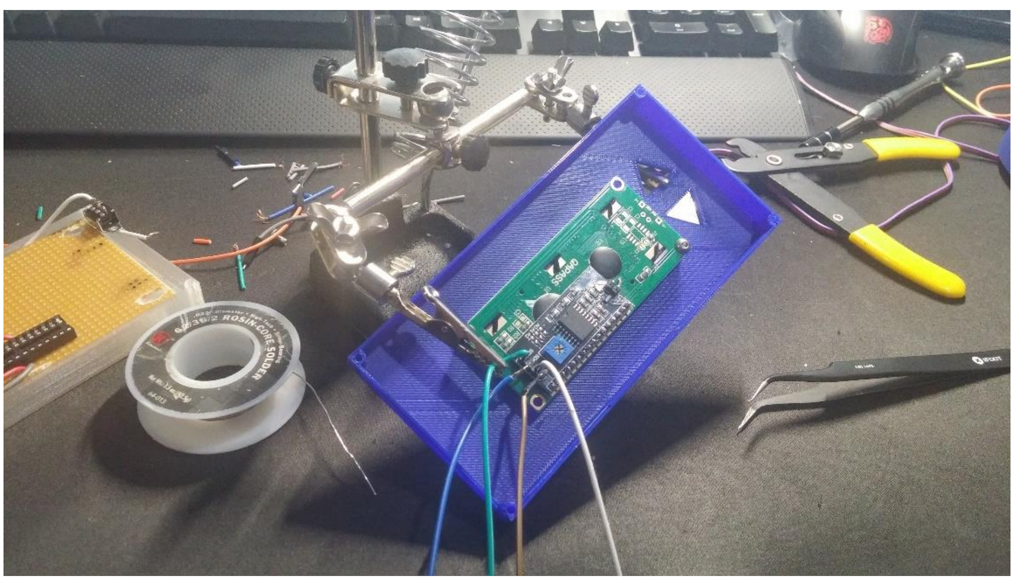

Figure 4. LCD Assembly

Figure 5. Final Assembly

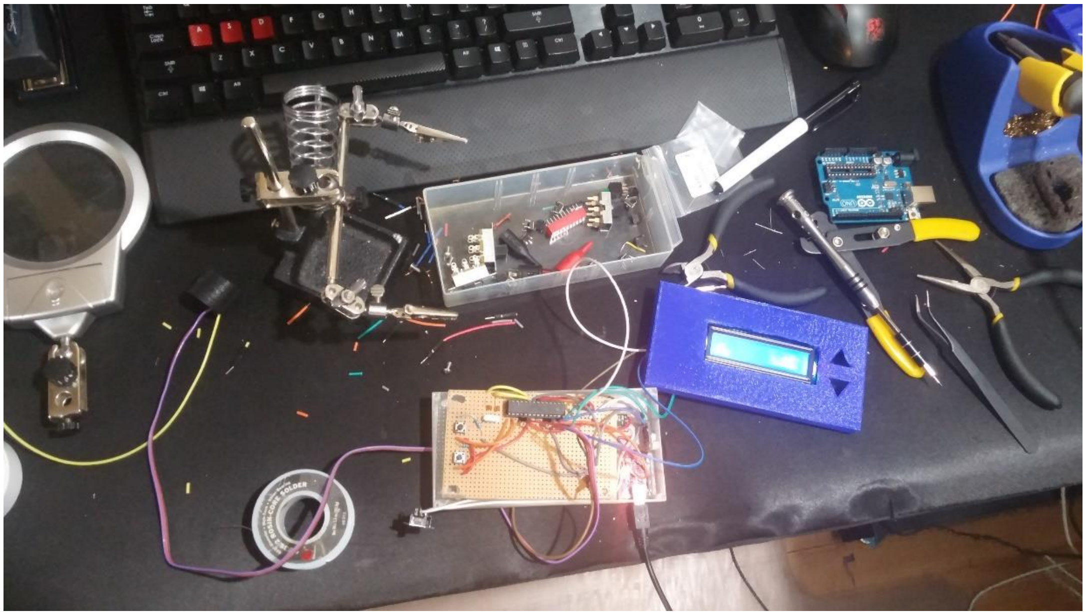

Figure 6. Initial Test on Battery Power

Results

Overall during testing the device functioned as desired and was able to take all measurements accurately and quickly at that. To be sure of this I compared the pulse sensor by first measuring my own heart rate with a stopwatch and then with the sensor. In the end the sensor was off by only +/- 1BPM compared to my own measurements and therefore is definitely a very accurate and precise device. Although, more testing would needed to be done using more proven methods than by using a stopwatch and counting. To test the heart rate sensor I compared it to a handheld IR thermometer. Comparing the readings of the two to the same objects resulted in differences of about +/- 0.5 °F across the board. Therefore, the temperature sensor as well is accurate and precise but again more scientific and rigorous testing would needed to be done to prove this. Finally, the battery charge meter was tested and compared to the readings of a multimeter. In order to calculate the percentage charge of the battery an analog pin on the Atmega 328p was connected to the positive lead of the battery and the voltage was measured. To calculate the percentage a mathematical function was created which relates the max voltage of the battery at full charge with what was considered the voltage of a dead battery. In this case, the max voltage of the LiPo battery was measured at 4.30V and the minimum voltage was measured at 3.70 V. Knowing these values the function below was created:

$$\boldsymbol{percentage} = \frac{(\boldsymbol{reading} - \mathbf{3}.70)}{(4.30 - 3.70)} * 100$$

From testing this function did accurately calculate the percentage of charge remaining on the battery. Although, this value had a tendency to fluctuate by about +/- 10% in real time on the LCD. This is likely due to the fact that an analog pin was being used to measure this voltage which have been shown to not be entirely accurately. Also, there was likely electrical noise interfering with the reading given that neither the analog pin nor the wire were shielded. Regardless, the function was still able to give an idea of the remaining charge on the battery which is definitely a useful and integral function, especially in a medical environment.

Discussion

Overall the HMBT functioned above expectations in reference to accuracy and reliability; frankly better than I expected. In the end the final cost was only about eight dollars off of my estimate of $115.50. This can be seen in detail in Figure 7. which is the final BOM for the total project. Currently, the HMBT is disassembled given that the voltage regulator was shorted out accidentally and subsequently destroyed therefore I have been unable to actually finish and test the device in the field and in more strenuous circumstances than around the house. Regardless though, I am confident in the HMBT’s ability and I am not concerned about more intense testing given how it has performed thus far. The project in total was very enjoyable and I learned a considerable amount along the way. I plan on continuing to improve the design by coming up with a more altruistic case design, designing and having a custom PCB fabricated, adding a larger capacity battery, and using a more colorful and larger display. Overall though, in it's current state I am very proud of the device and what it is capable of and more so what I have accomplished.

Part	Quantity	Price
Atmega 328p Chip	1	$1.63
16x2 LCD	1	$5.98
Pulse Sensor	1	$34.68
Temperature Sensor	1	$6.13
2000mAh Li-Po Battery	1	$24.61
5V Step up Board	1	$8.90
Micro USB Li-Po Charger	1	$4.57
3D Printer Filament	800 grams	$28/kg 28 * .800kg = $22.40
Assorted Misc. Electronics	1	$15.00
		Total: $123.90

Figure 7. Final Bill-of-Materials