Pulseband: *A Hands-on Tutorial on How to Design a Smart Wristband to Monitor Heart-Rate*

Nicholas Constant, Tanya Wang, Kunal Mankodiya
Department of Electrical, Computer, and Biomedical Engineering
University of Rhode Island, Kingston, RI, USA
n.constant.2014@ieee.org, tanya_wang@uri.edu, kunalm@uri.edu

***Abstract*— Wearable devices like smartwatches are becoming an accepted way to monitor a person's health stats such a heartrate and physical activity. This paper will explain how to develop a do-it-yourself (DIY), smart wristband to perform photoplethysmography (PPG), also known as pulse oximetry. The designed Pulseband acquires PPG signals and estimates heart rate (HR) in real-time. HR estimates can be transmitted via a Bluetooth Low Energy (BLE) to a nearby placed smartphone or computer. Specifically, we will delve into three key aspects essential to the band; hardware, software, and 3D printing.**

Keywords—PPG; BLE; mobile health; wearable sensors, Internet of things; Smart Wristband.

I. INTRODUCTION

This paper describes the methods used to build a pulse sensing wristband called *Pulseband*. The band itself consists of only three main components making it a low-cost, low-power, and easily customizable device tailored for the DIY community. Its small size means that the maker can easily change the measurement location.

The *Pulseband* is a wristband that can realize the user's heartbeat, measured in beats per minute (BPM), through the combined use of a photoplethysmographic (PPG) sensor and a microprocessor. The microprocessor transmits the heart rate via a BLE connection. This entire system consists of only 3 essential components; a PPG sensor, microprocessor with an embedded BLE radio, and a rechargeable 3.6V lithium-ion coin cell battery. The ability to have very few components involved means that it can be entirely packaged within a 1.25"x1.25"x.75" 3D printed case. We placed the casing inside a 3D printed wristband to mimic the readings used by a smart watch.

II. MATERIALS & METHODS

A. PPG Sensor

The PPG sensor measures pulse oximetry by means of a noninvasive and reflective photo-based technique [1]. The technique places a light and photodetector on the surface of the skin. The light transmits through the skin and blood but is largely reflected by the bone thereby traveling back to the photodetector. As the light journeys through the skin, the photodetector monitors fluctuations in light absorption caused by the variations in blood volume passing through the arteries. This allows us to calculate the duration of the cardiac cycle, also known as heart rate, using the peaks of light absorption [2]. The PPG sensor uses a green 565nm LED and photodetector with a high sensitivity for the same wavelength. We choose a green light because its dominant absorption corresponds to the red blood cells.

While the photodetector is the most sensitive to the wavelength of light used by the LED; it can still detect an unwanted wavelength of light if it is bright enough. The unwanted light might come from the lights used within a room or even come from the natural light outdoors. We can reduce the amount of noise passed from the photodetector to the microprocessor by inserting a low pass filter between the two. We choose a low pass filter because the unwanted noise such as motion artifacts is found at higher frequencies, and eliminating the high frequencies provides clearer information about heart rate.

The final piece of the PPG sensor is an op-amp. The filtered signal is amplified by the op-amp, before reaching the microprocessor. The op-amp with high input impedance is preferred in order to avoid electromagnetic interference. Furthermore, the output impedance of the op-amp is so low that is does not weigh down on the microprocessor.

B. Signal Processing

The PPG sensor outputs an analog signal ranging from zero to three volts based on the intensity of light seen by the photodetector. This signal is processed by the microprocessor which performs an adaptive peak detection

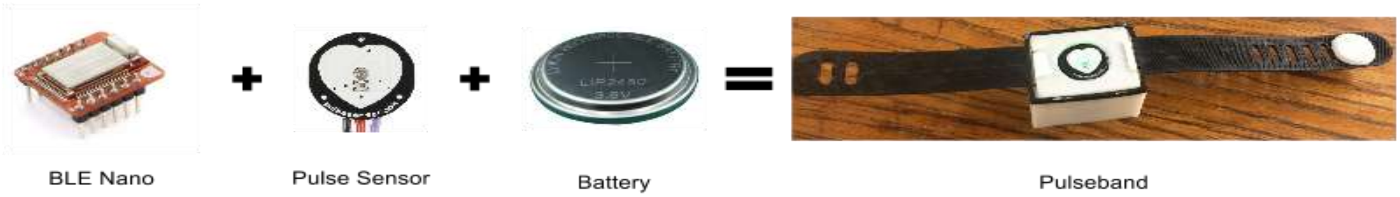

Fig. 1. Overview of Pulseband

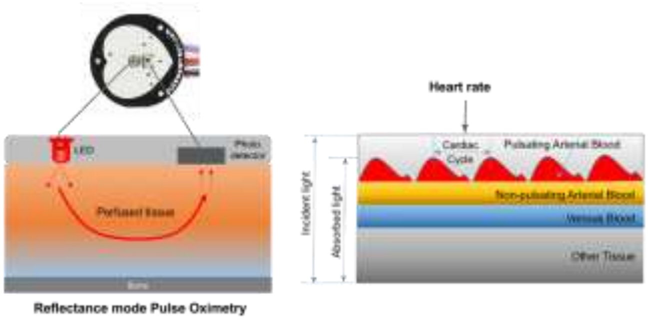

Fig. 2. Breakdown of Reflective Pulse Oximetry

algorithm. The algorithm starts by normalizing the signal to a range of 0-512 and sets the peak detection threshold to 256. After the threshold is crossed for the first time, the actual peak is stored followed by the corresponding trough. The new threshold is set as the midpoint between both the peak and trough.

Once the first beat has been detected, a timer will run for up to 2 seconds. If a peak is not detected within that time frame then the next beat will be considered the first. Otherwise, when a second beat is detected the time passed between consecutive beats is stored in a first in, first out (FIFO) array. Once this array is filled it averages out the stored times and use 60 over that average to calculate the beats per minute (BPM) [3].

C. Transmitting data

The calculated BPMs are then sent over an available BLE connection using the standardized GATT Heart Rate Service. This service provides a few informative characteristics about the detecting device such as heart rate values, sensor location, and control points [4]. The heart rate value is labeled as hrmCounter in the code and is set as an unsigned 8-bit integer. The instance of the GATT Heart Rate Service can be updated by calling the object's function updateHeartRate. This function is defined in the GATT Heart Rate Service header. Updating objects through its functions is the standard method used by the BLE libraries. Once the object is updated the changes will be transmitted to the paired device.

The *Pulseband* is always updating its local heart rate values without regard to a connected device, because it is handled by an interrupt. However, it will only update the instance of the GATT Heart Rate Service once the band is paired with a BLE-enabled device.

III. ASSEMBLY

The *Pulseband* is designed to be easily assembled. The process can be broken down into three sections, programming the board, 3D printing the housing, and combining the components.

A. Programming the Pulseband

The BLE Nano can initially be programmed using the mbed compiler along with the MK20 USB board. The complete set of code can be found on the mbed website [5]. The code is well commented and ready to be successfully

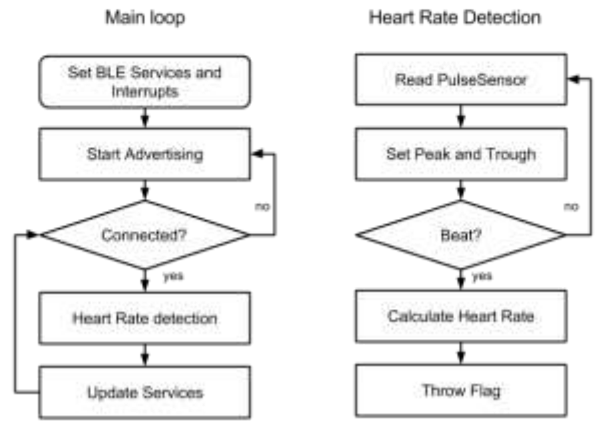

Fig. 3. Pulseband program flowchart

compiled for the BLE Nano. When the code compiles, a .hex file will be automatically generated and downloaded through your browser. This hex file contains everything the BLE Nano requires to run the program.

In order to upload the program to the BLE Nano, the MK20 USB board must initially be used as the interface to the computer. After installing the appropriate drivers, the BLE Nano will appear as a disk drive labeled “MBED”. The hex file can be uploaded by copying it to the drive. The BLE Nano restarts itself when going through the uploading process and does not display the hex file after a successful upload. As a final note, the board can be set up to allow programming through BLE and the Nordic Semiconductor Utility smartphone app, which will not be discussed in this paper.

B. 3D Printing

The *Pulseband* project enclosure consists of two separate parts, a rigid encapsulating housing that provides structure and houses all the components, and a flexible adjustable band that allows the device to be worn by a user. Both of these parts are fabricated using Fused Filament Fabrication (FFF) 3D printing, in our case with the Makerbot Replicator 2x. The flexible band is made from NinjaFlex’s SemiFlex, a thermoplastic elastomer (TPE), while the rigid housing is made from the Makerbot brand Acrylonitrile Butadiene Styrene (ABS) plastic. Each individual part is sketched in Solidworks, and careful planning is required to design parts that fit the internal electronics, the interface between the two materials, as well as the user. The SolidWorks parts should be saved in .STL files and opened in Makerware for 3D printing. There is an allowable tolerance for misprints, and the final design usually requires a bit of trial and error.

The final design used for this product consists of a box, with a bottom that is fused on with ABS slurry, a mixture of extra ABS plastic dissolved in an equal part of acetone. The printing of the ABS parts requires the use of a heated build plate (110℃) covered in kapton tape, and the extruder nozzle temp to be set to around 225℃. Catastrophic failure may occur when the part does not

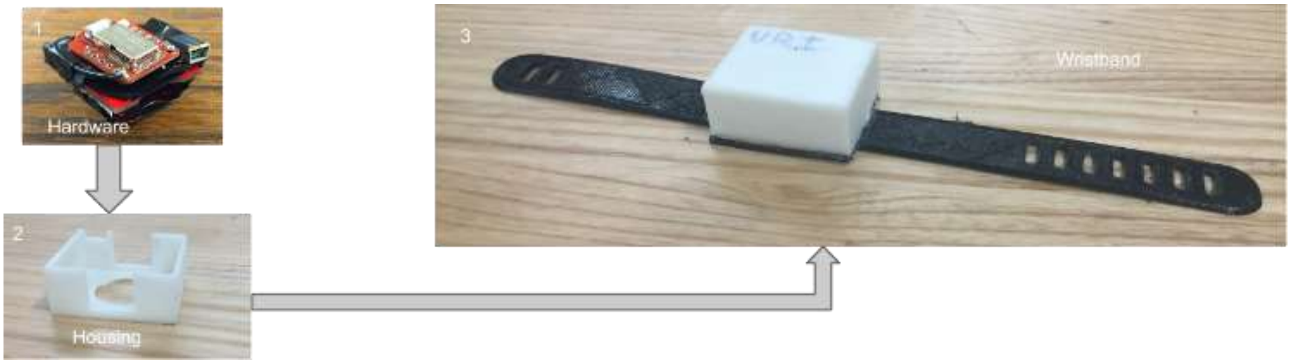

Fig. 4. Setting up the Pulseband

adhere to the build plate, and will cause the part to warp, or even completely lift off the plate. Some simple solutions to this problem is to use helper discs, a feature built right into the MakerWare software, or our preferred method of applying a very thin layer of ABS slurry to the build plate before printing. The flexible filament does require more finesse and care to achieve successful parts, but when done right can give a product a very clean and professional finish. This requires the extruder temperature can be dropped to 215℃, and does not require a heated build plate, although ours is set to 40℃ during this process. Adhesion to the build plate requires painters tape instead of kapton, but no other prep-work or slurry is needed. For more elaborate finishes, the Replicator 2x has a dual-extruder which can create dual-material and/or dual-colored objects. The major cause for failure of a flexible filament is an increased risk of buckling when the extruder pushes non-rigid filament through the nozzle and causes the nozzle to clog. The general solution to this problem is to modify the drive block to a spring-loaded 3D printed replacement, available on the Makerbot's Thingiverse open source library. The last piece required is a small ABS two-prong clip that fastens the band together.

C. Combining components

At this point, the BLE Nano no longer requires the MK20 USB board, and all header pins can be removed. The PulseSensor does not require any more than an inch and a half of wire, and can be connected to VDD, GND (either ground is fine) and P-06. The coin cell battery can use the same ground as the PulseSensor and its positive terminal can us VIN. Before installing the battery, a piece of electrical tape can be used to isolate the different components from shorting with one another. Once the each board is protected they can be packed together to fit within the 3D printed housing. The PulseSensor can fit flush within the hole at the bottom of the housing and the battery should be easily accessible as this design requires the battery to be removed to power down. The housing filled with the *Pulseband* hardware can slide tightly into the wristband. Once the *Pulseband* is ready for use it will start advertising immediately. It can now be tested using any BLE scanner application. It is best to find an application, such as PanoBike to the test the *Pulseband* since this application is ready to interact with a BLE device using the GATT Heartrate service.

IV. FUTURE DEVELOPMENTS

The *Pulseband* can be viewed as a very basic building on which a variety of other applications can be integrated. All of the code from this project can be learned from and built up to incorporate more GATT services and control a more complex set of sensors. The BLE Nano has SPI and I2C capabilities which makes integrating the ability to communicate with other boards a quick process. Lastly the 3D models can be altered to create a customized design for better fit and style.

Acknowledgements

This work is partially funded by the University of Rhode Island Research Council.

REFERENCES

[1] Tamura, T. Wearable Photoplethysmographic Sensors—Past and Present. Electronics (Basel) 3.2 Jun 2014: 282-302. MDPI AG. 02 Sep 2015.

[2] Torres, D. "Build A Wrist Heart-Rate Monitor Using An Ultra-Low-Power MCU." Build A Wrist Heart-Rate Monitor Using An Ultra-Low-Power MCU. 17 July 2013. Web. 17 Aug. 2015.

[3] Constant, N., Douglas-Prawl, O., Johnson, S., and Mankodiya, K. "Pulse-Glasses: An Unobtrusive, Wearable HR Monitor with Internet-of-Things Functionality." In Wearable and Implantable Body Sensor Networks (BSN), 2015 12th International Conference on, IEEE, 2015.

[4] "Tutorial 2: Heart Rate Monitor (BLE Services)." Introduction to Mbed BLE. Web. 2 Aug. 2015.

[5] https://developer.mbed.org/users/kabuki4774/code/NANO_HeartRate/