Heart Rate Monitor Project Details

Group 6

Erick Blandon Ramirez

Oscar Bonilla

Sandor Souvannakoumane

# Table of Contents

Contents

[Table of Contents 2](#_Toc51513759)

[Hardware 3](#_Toc51513760)

[Parts List 3](#_Toc51513761)

[Sensors 3](#_Toc51513762)

[Controllers 3](#_Toc51513763)

[Communications 3](#_Toc51513764)

[Circuit 3](#_Toc51513765)

[Software 3](#_Toc51513766)

[Language 3](#_Toc51513767)

[Configurations Control 3](#_Toc51513768)

[UI Requirements 3](#_Toc51513769)

[Backend Requirements 4](#_Toc51513770)

# Hardware

## Parts List

* Arduino Uno (<https://store.arduino.cc/usa/arduino-uno-rev3> )
* Pulse Sensor (<https://pulsesensor.com/>)
* HiLetgo HC-05 Wireless Bluetooth RF Transceiver (<http://www.hiletgo.com/ProductDetail/2152062.html> )
* Android device

## Sensors

The pulse sensor is the only sensor we plan on using. It can detect a pulse and send a corresponding analog signal. This signal can then be used for further analysis and assessment with our program.

## Controllers

We are using an Arduino Uno as our controller. It is capable of handling the output signal of the pulse sensor and using a Bluetooth module to communicate with an Android device. The Arduino Uno was chosen for convenience at the time of development. In a production scenario, an Arduino Nano or smaller microcontroller would be more convenient for the user and cost effective.

## Communications

The HiLetgo HC-05 Wireless Bluetooth RF Transceiver is the planned part for use in the project, though others could work as well. This transceiver is capable of connecting to an android device and sending information. This will allow us to send live data to the phone for processing and our GUI.

## Circuit

To be created in Fritzing or a similar program.

# Software

## Language

The Arduino Uno will use its C based language for development and control of the pulse sensor and Bluetooth module. The final code will be stored directly on the Arduino Uno, allowing for portability.

The Android program has yet to have the details determined. The programming language will be determined at a later time. Likely to be in C# or similar.

## Configurations Control

We will be using a Github repository (<https://github.com/emb-cpp/Heart-Rate-Monitor> ) that holds our code for both the Arduino and Android application.

## UI Requirements

Currently, our GUI must be able to display a live updating BPM, a pulse graph, a trending report, and a live deviation from the average. Other features can be discussed in future development.

## Backend Requirements

The Android program must be able to connect to the Bluetooth transceiver to receive information. It must be able to process the information BPM information and create a historical average. It must be able to calculate the deviation from the average for the user. Other requirements can be discussed in future development.