

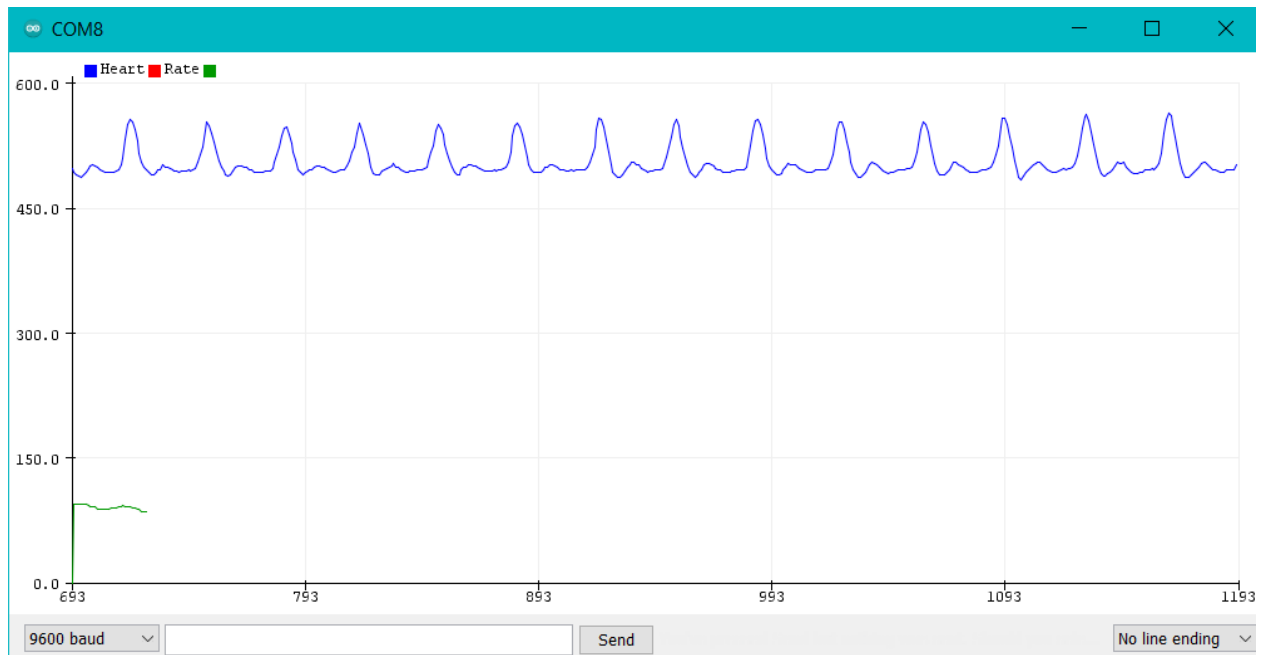
Project Description:

The arduino microcontroller was connected to an LED and buzzer, which turned on in sync with the heart rate of the user. The code can be modified to change the frequency of the buzzer and the serial plotter shows a real-time EKG graph of signal strength over time.

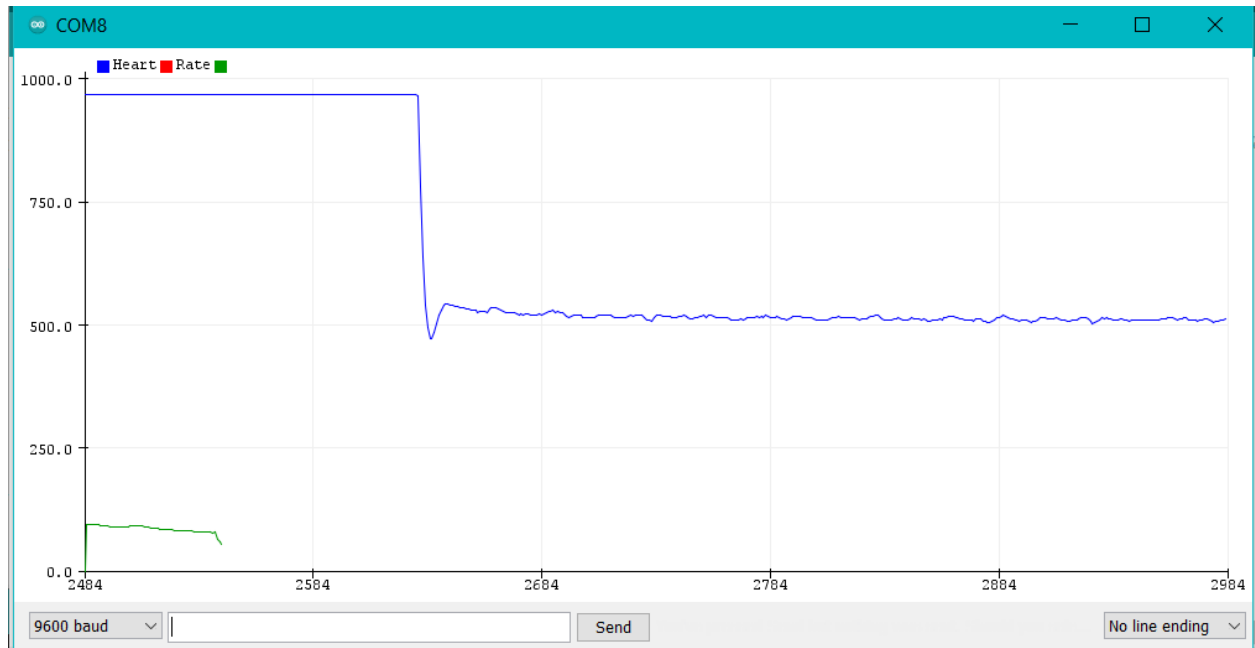
Sample Data

Graph of EKG:

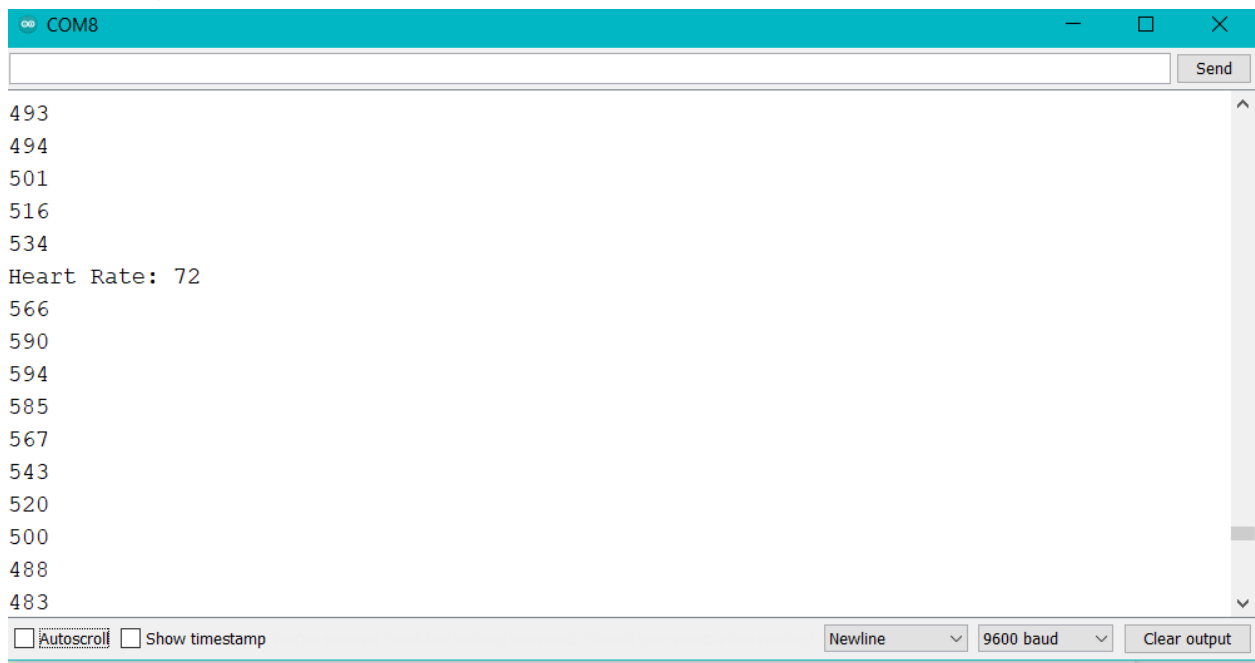
Heart rate over time shown in green and pulse signals shown in blue



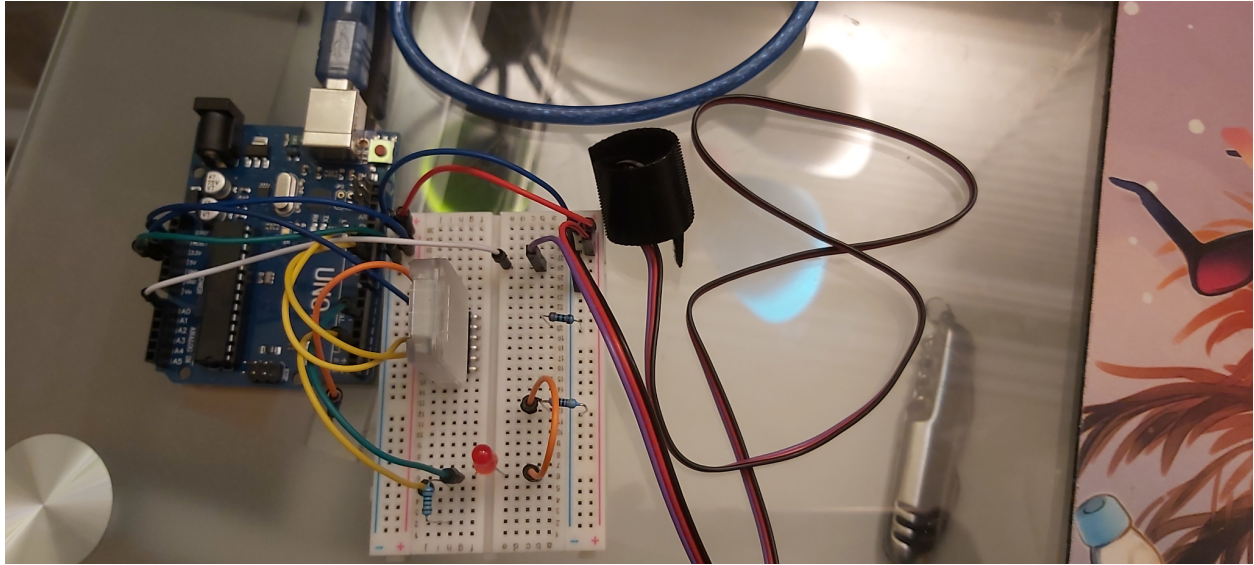
HR Flatline Signal:



Serial Monitor for Printed Signal Values and Heart Rates (threshold for heart beat set to 550)



Arduino Setup / Circuit Layout:



Arduino and breadboard with bluetooth module and HR sensor with analog output signal

C Code: compiled in Arduino IDE

```
/* PulseSensor Starter Project and Signal Tester
 * The Best Way to Get Started With, or See the Raw Signal of, your PulseSensor.com™ &
Arduino.
```

```
*
```

```
* Here is a link to the tutorial
```

```
* https://pulsesensor.com/pages/code-and-guide
```

```
*
```

```
* WATCH ME (Tutorial Video):
```

```
* https://www.youtube.com/watch?v=RbB8NSRa5X4
```

```
*
```

```
*
```

-
- 1) This shows a live human Heartbeat Pulse.
 - 2) Live visualization in Arduino's Cool "Serial Plotter".
 - 3) Blink an LED on each Heartbeat.
 - 4) This is the direct Pulse Sensor's Signal.
 - 5) A great first-step in troubleshooting your circuit and connections.
 - 6) "Human-readable" code that is newbie friendly."

```
*/
```

```
#define USE_ARDUINO_INTERRUPTS true
```

```

#include <PulseSensorPlayground.h>

// Variables
int PulseSensorPurplePin = A0;    // Pulse Sensor PURPLE WIRE connected to ANALOG PIN
0
int LED13 = 13; // The on-board Arduino LED

int Signal;           // holds the incoming raw data. Signal value can range from 0-1024
int Threshold = 550;  // Determine which Signal to "count as a beat", and which to ignore.
const int PIN_SPEAKER = 2; // speaker on pin2 makes a beep with heartbeat
const int PULSE_BLINK = 5; // Pin 13 is the on-board LED
const int PULSE_FADE = 5;

PulseSensorPlayground pulseSensor;

// The SetUp Function:
void setup() {
  pinMode(LED13,OUTPUT);    // pin that will blink to your heartbeat!
  Serial.begin(9600);      // Set's up Serial Communication at certain speed.

  pulseSensor.analogInput(PulseSensorPurplePin);
  //pulseSensor.blinkOnPulse(LED13);    //auto-magically blink Arduino's LED with heartbeat.
  pulseSensor.setThreshold(Threshold);

  pulseSensor.blinkOnPulse(PULSE_BLINK);
  pulseSensor.fadeOnPulse(PULSE_FADE);
}

// The Main Loop Function
void loop() {

  Signal = analogRead(PulseSensorPurplePin); // Read the PulseSensor's value.
  // Assign this value to the "Signal" variable.
  int myBPM = pulseSensor.getBeatsPerMinute();

  Serial.println(Signal);          // Send the Signal value to Serial Plotter.

  if(Signal > Threshold){          // If the signal is above "550", then "turn-on" Arduino's
on-Board LED.
    digitalWrite(LED13,HIGH);
  } else {

```

```

    digitalWrite(LED13,LOW);          // Else, the signal must be below "550", so "turn-off" this
LED.
}

if (pulseSensor.sawStartOfBeat()) {    // Constantly test to see if "a beat happened".
    pulseSensor.outputBeat();
    Serial.println("♥ A HeartBeat Happened ! "); // If test is "true", print a message "a heartbeat
happened".
    Serial.print("BPM: ");              // Print phrase "BPM: "
    Serial.println(myBPM);              // Print the value inside of myBPM.
    tone(PIN_SPEAKER,1047);

}

if(pulseSensor.isInsideBeat() == false){
    noTone(PIN_SPEAKER);
}

delay(10);
}

```