Overview – Heart Pulse Sensor with IBI display and fast audio replay

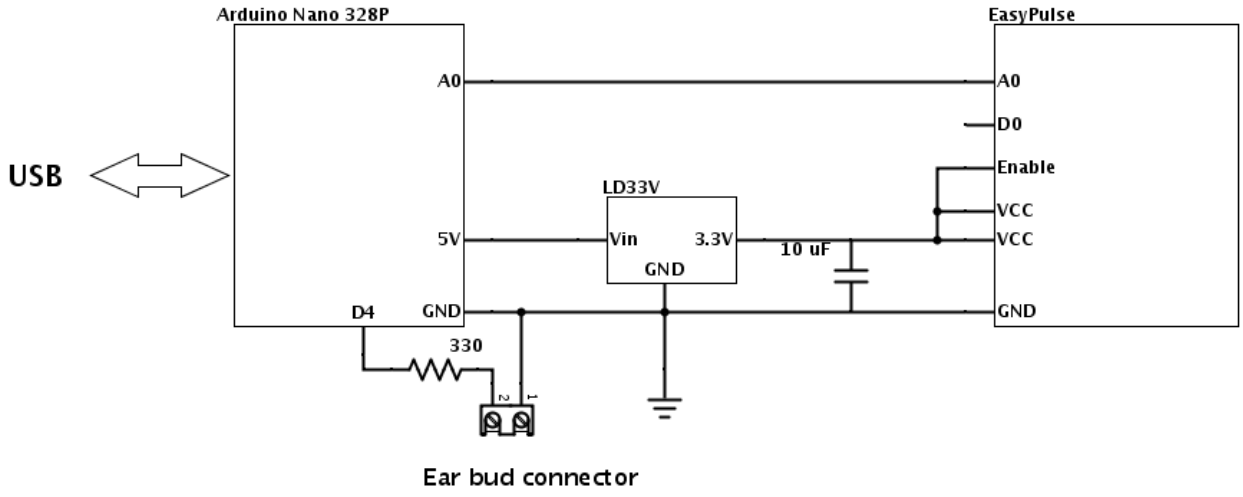
My objective was to make a pulse sensor that can display the statistical distribution of the interbeat interval (IBI) of a reasonably large number of heart pulses and allow audio playback of a reasonably long series of beat pulses at an accelerated speed. Fast audio playback allows the ear to hear the heart as a drumbeat and detect beat patterns that may exist.

Hardware

- Arduino Nano with AtMega328 microcontroller
- EasyPulse Sensor v1.1 (www.embedded-lab.com)
- LD33V 3.3 volt linear regulator
- 10 uF ceramic capacitor
- 330 ohm resistor
- Single ear bud (salvaged from an airline handout).

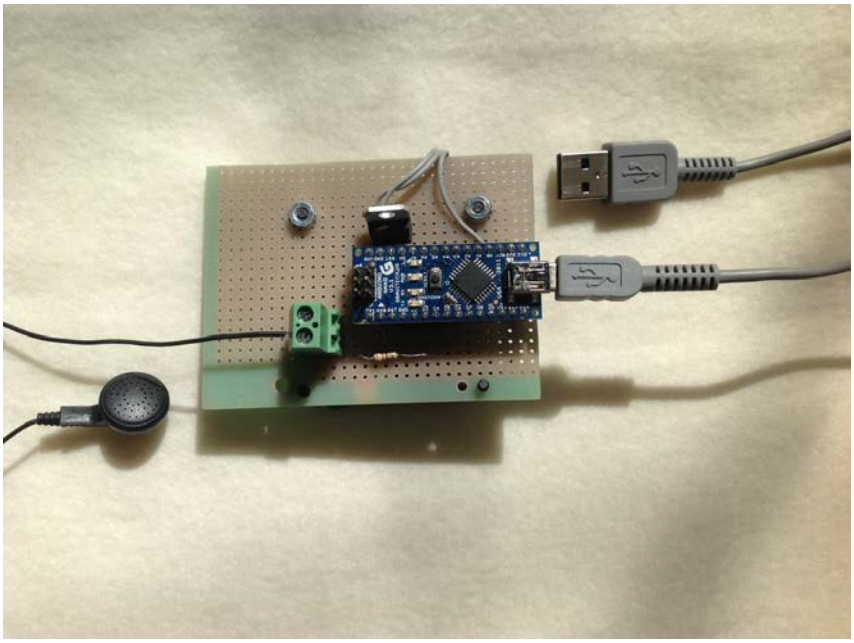
The EasyPulse can be directly powered by the 5V output of the Nano, but we have found that noise from this USB 5v source makes the EasyPulse output almost unusable. The main purpose of the 3.3v regulator is to provide the EasyPulse with a clean supply voltage.

Circuit schematic

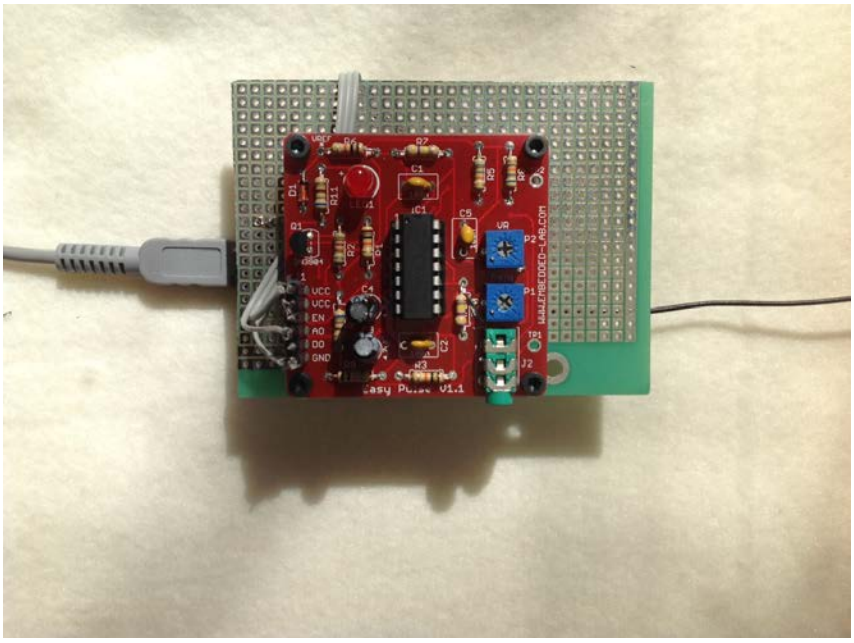


Assembly

The Arduino Nano and the EasyPulse sensor were mounted on opposite sides of a perforated proto board.



Top view showing Arduino Nano and ear bud connector



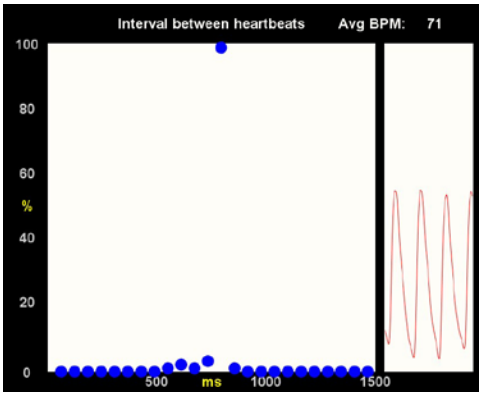
Bottom view showing EasyPulse v1.1 board.

Software

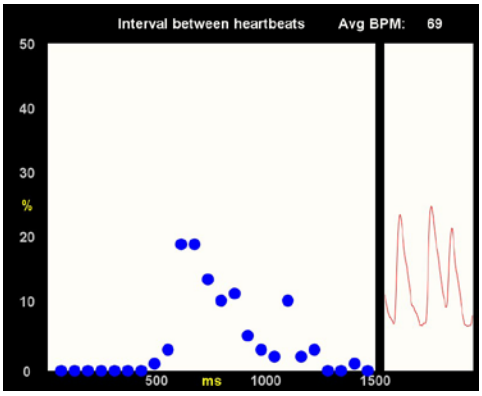
Software for the project was based on Arduino and Processing code developed by the folks at www.pulsesensor.com and posted at https://code.google.com/p/pulse-sensor/. We have modified the Arduino code to allow the Arduino to play back a drumbeat-like audio sequence based on a saved sequence of 100 heartbeats. The beat rate is accelerated by a factor of 8 to more easily allow the ear to hear beat any patterns that may exist. The Processing code has been modified to display a plot of the statistical distribution of interbeat intervals for 100 beats. The Arduino and Processing code is easily modified to allow collection of more or less than 100 beats.

During operation, the Arduino uses the "PulseSensor_FastReplay" sketch to sample the filtered and amplified analog output of the EasyPulse sensor, store 100 sequential pulses in an array, and send the signal amplitude and interbeat interval to the serial port.

The Processing program "PulseSensor_IBIdistribution" receives the serial input and generates a plot of the time distribution of the interbeat intervals (IBI) , as well as the pulse signal and the average heart rate (displayed after 100 pulses). IBI plots captured for a normal heart rhythm and a heart arrhythmia are shown below.



IBI distribution with normal (sinus) rhythm



IBI distribution with arrhythmia. Note change in vertical scale.

After 100 pulses, the program also displays the average beats/minute.

The Processing program uses several keyboard inputs to reset the data series, save the display, change the vertical scale of the display, and start fast audio replay of the beat series.

Key ‘C’ clears the 100 point data array in both Processing and the Arduino sketch

‘S’ saves a .jpg of the display to the Processing sketch folder with a title in the format MonthDay-HourMinute.

‘1’ sets the vertical display scale 0 – 100%.

‘2’ sets the vertical display 0 – 50%.

‘F’ starts the playback of the audio signal at the ear bud. The signal is a series of 100 clicks played at 8X the speed of the original heartbeat pattern. (Interclick interval is 1/8 the IBI). The same series will be replayed each time ‘F’ is pressed. Press ‘C’ to collect a new data series for display and replay.

Please be aware that the first few data points collected immediately after the program is started or after the sensor is repositioned on the finger are usually invalid.