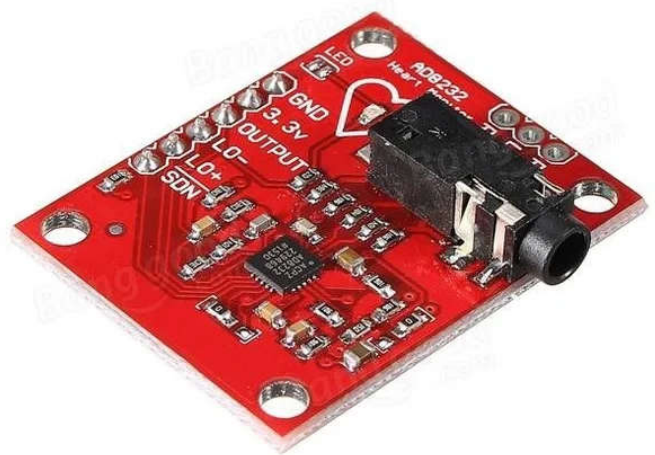


Heart Rate Monitor AD8232, Arduino, Processing

By [stevensarns \(/member/stevensarns/\)](/member/stevensarns/) in [Circuits \(/circuits/\)](/circuits/) > [Arduino \(/circuits/arduino/projects/\)](/circuits/arduino/projects/)

10.280 10 1



[\(/member/stevensarns/\)](/member/stevensarns/)

By **stevensarns**
[\(/member/stevensarns/\)](/member/stevensarns/)

About: Retired embedded system programmer and hardware designer. [More About stevensarns » \(/member/stevensarns/\)](#)

More by
the author:



The Analog Devices AD8232 is a complete analog front end designed to acquire milliVolt level EKG (ElectroCardioGram) signals. Although it is a simple matter to hook up the AD8232 and see the resulting EKG signal on an oscilloscope, the challenge for me was to acquire the signal for display on my PC. That is when I discovered Processing!

AD8232 documentation page - <http://www.analog.com/en/products/application-spec...> (<http://www.analog.com/en/products/application-specific/medical/ecg/ad8232.html>)

A breakout board is available from Sparkfun here - <https://www.sparkfun.com/products/12650> (<https://www.sparkfun.com/products/12650>) or, if you wait a

few weeks, from China here - <https://www.ebay.com/itm/New-Single-Lead-AD8232-Pu...> (https://www.ebay.com/itm/New-Single-Lead-AD8232-Pulse-Heart-Rate-Monitor-ECG-Kit-Module-For-Arduino/262752356143?ssPageName=STRK%3AMEBIDX%3AIT&_trk=sid=p2057872.m2749.l2649).

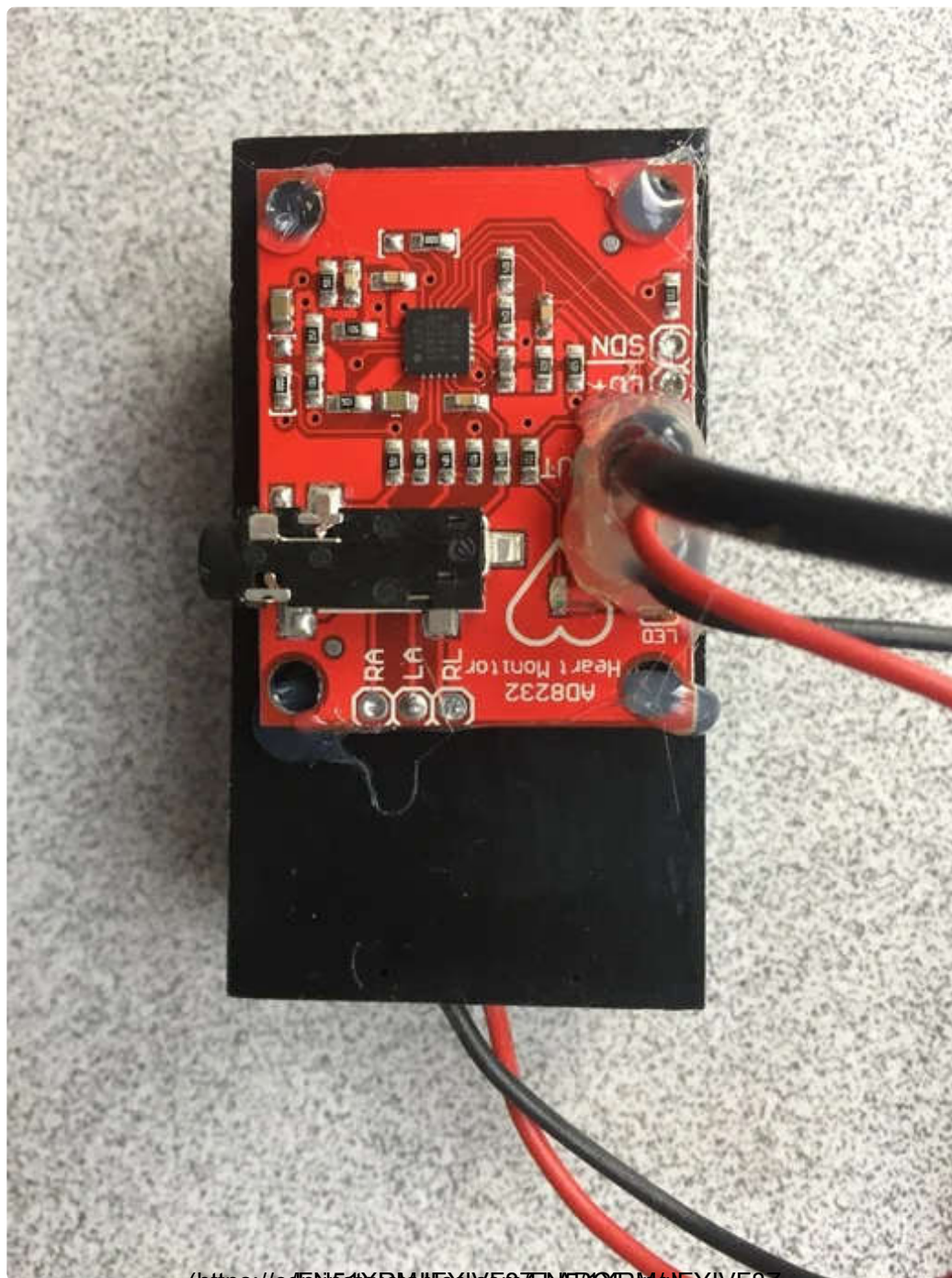
I ordered the kit including body sensor cable with sticky pads.

Teacher Notes

Teachers! Did you use this instructable in your classroom?

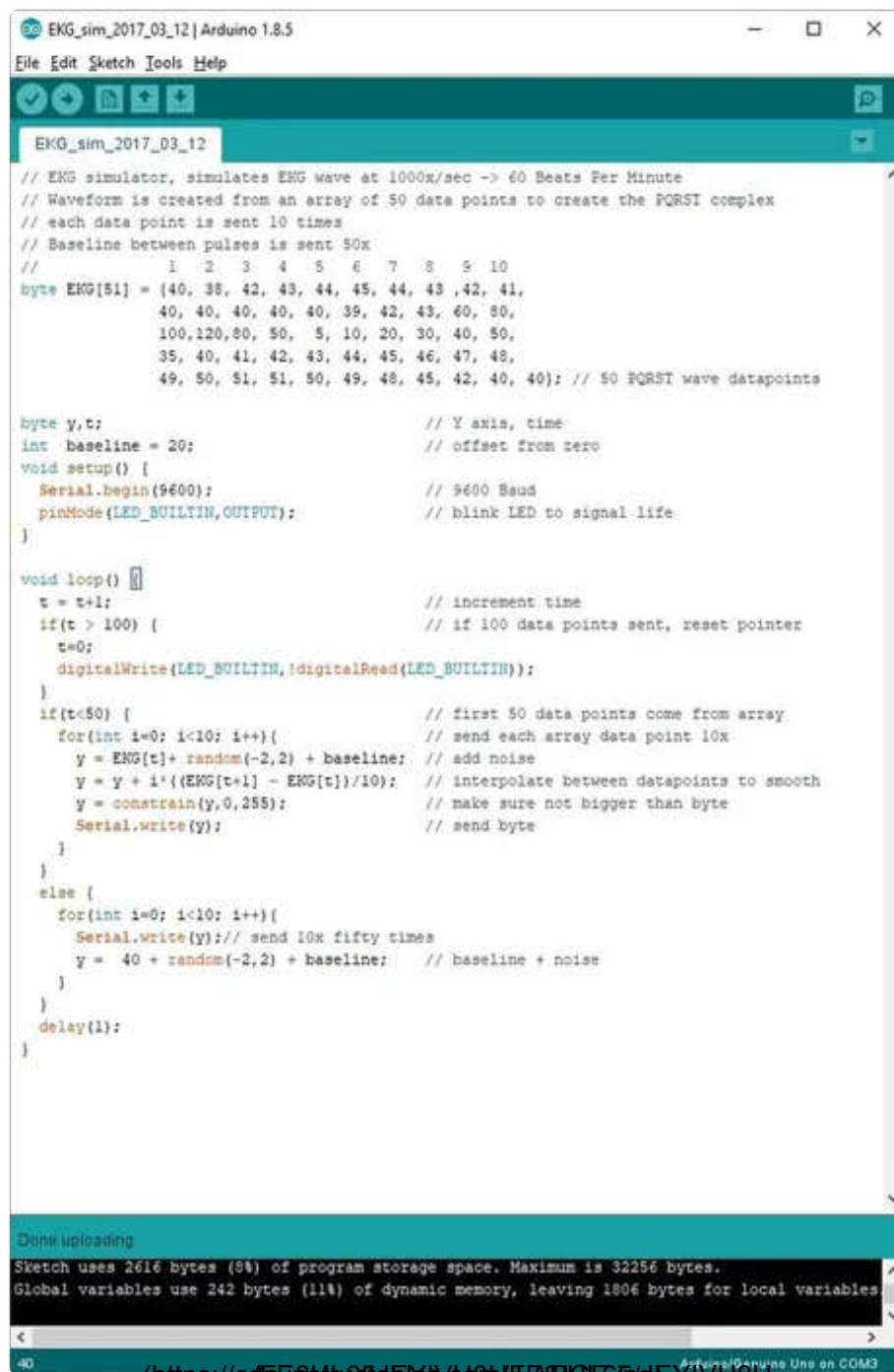
Add a Teacher Note to share how you incorporated it into your lesson.

Step 1: Preparing the AD8232 Breakout Board



The plan is to have the AD8232 board acquire the EKG signal. The output of the AD8232 is a signal of approximately 1.5 Volts. This signal will be sampled by an Arduino Uno at approximately 1k samples/second. These samples values are then sent over the USB port to the PC for display. I quickly discovered that powering the AD8232 from the 3.3V output of the Arduino board was a bad idea - too much 60 Hz noise. So I switched to 2 x AA batteries. The AD8232 can be powered by a 3V mercury coin cell if desired. Two wires (signal and ground) ran from the AD8232 board to the Arduino (A0 and ground). I used a generous amount of hot melt glue to reinforce the wires at the the AD8232 board junction.

Step 2: EKG Simulation on Arduino Uno



```
EKG_sim_2017_03_12 | Arduino 1.8.5
File Edit Sketch Tools Help

EKG_sim_2017_03_12

// EKG simulator, simulates EKG wave at 1000x/sec -> 60 Beats Per Minute
// Waveform is created from an array of 50 data points to create the PQRS complex
// each data point is sent 10 times
// Baseline between pulses is sent 50x
//
//      1  2  3  4  5  6  7  8  9 10
byte EKG[51] = {40, 38, 42, 43, 44, 45, 44, 43, 42, 41,
               40, 40, 40, 40, 40, 39, 42, 43, 40, 38,
               100,120,80, 50,  5, 10, 20, 30, 40, 50,
               35, 40, 41, 42, 43, 44, 45, 46, 47, 48,
               49, 50, 51, 51, 50, 49, 48, 45, 42, 40, 40}; // 50 PQRS wave datapoints

byte y,t;
int baseline = 20; // offset from zero
void setup() {
  Serial.begin(9600); // 9600 Baud
  pinMode(LED_BUILTIN,OUTPUT); // blink LED to signal life
}

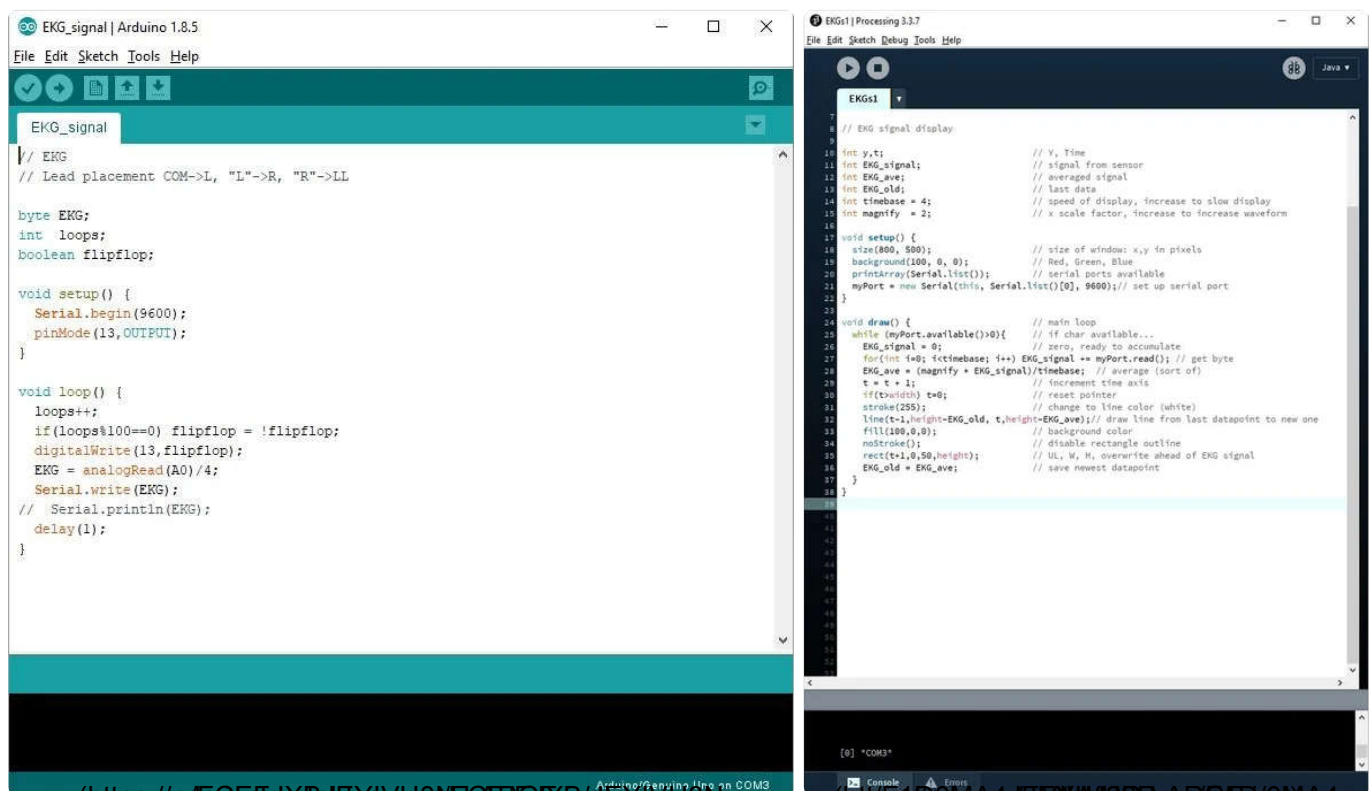
void loop() {
  t = t+1; // increment time
  if(t > 100) { // if 100 data points sent, reset pointer
    t=0;
    digitalWrite(LED_BUILTIN,!digitalRead(LED_BUILTIN));
  }
  if(t<50) { // first 50 data points come from array
    for(int i=0; i<10; i++){ // send each array data point 10x
      y = EKG[t]+ random(-2,2) + baseline; // add noise
      y = y + i*{(EKG[t+1] - EKG[t])/10}; // interpolate between datapoints to smooth
      y = constrain(y,0,255); // make sure not bigger than byte
      Serial.write(y); // send byte
    }
  }
  else {
    for(int i=0; i<10; i++){
      Serial.write(y); // send 10x fifty times
      y = 40 + random(-2,2) + baseline; // baseline + noise
    }
  }
  delay(1);
}

Done uploading
Sketch uses 2616 bytes (8%) of program storage space. Maximum is 32256 bytes.
Global variables use 242 bytes (11%) of dynamic memory, leaving 1806 bytes for local variables

40 (https://www.instructables.com/id/Heart-Rate-Monitor-AD8232-Ardui...)
```

The next step is to create a simulator running on the Arduino. This way I don't have to sit around with the electrodes attached to my body as I am debugging code.

Step 3: Up and Running



Finally, the PC display. The Arduino code needs to be changed to acquire real data instead of simulation data. The Processing code is shown. I was rather apprehensive about diving into a new language / development environment, but as soon as I saw the Processing IDE I thought "Whoa! This looks familiar - just like the Arduino." Here is the download link for [Processing \(https://processing.org/download/\)](https://processing.org/download/). It only took a few hours of hacking code I found on the Internet to get an application up and running. I discovered that the placement of the 3 electrodes on my body did not correspond to the notations on the wires. In my

case, the lead marked "COM" goes to the left, "L" goes to the right and "R" goes to the left leg.

My approach was to program the Arduino to acquire the signal and transmit it to the Processing application running on the PC. There may be another way; use Processing to directly control the Arduino - [link \(https://playground.arduino.cc/Interfacing/Processing\)](https://playground.arduino.cc/Interfacing/Processing). Even better, it may be possible to eliminate the Arduino altogether and use the PC audio port to acquire the signal through Processing - see this [Instructable \(https://www.instructables.com/id/DIY-EEG-and-ECG-Circuit/\)](https://www.instructables.com/id/DIY-EEG-and-ECG-Circuit/).

Step 4:

Here are the source files for the Arduino simulator, Arduino signal acquisition and Processing signal display.



EKG_signal.ino

(<https://cdn.instructables.com/ORIG/FYA0LB0/JF349CUQ/FYA0LB0JF349CUQ.ino>)



EKG_sim_2017_03_11....

(<https://cdn.instructables.com/ORIG/FFN/6NPO/JF349CUT/FFN6NPOJF349CUT.ino>)



EKG_simple.pde

(<https://cdn.instructables.com/ORIG/FG3/6ZK0/JF349CUX/FG36ZK0JF349CUX.pde>)
