ADVANCED MATERIALS

Supporting Information

for *Adv. Mater.*, DOI: 10.1002/adma.201701985

Wearable Microfluidic Diaphragm Pressure Sensor for Health and Tactile Touch Monitoring

*Yuji Gao, Hiroki Ota, Ethan W. Schaler, Kevin Chen, Allan Zhao, Wei Gao, Hossain M. Fahad, Yonggang Leng, Anzong Zheng, Furui Xiong, Chuchu Zhang, Li-Chia Tai, Peida Zhao, Ronald S. Fearing, and Ali Javey**

Supporting Information

Wearable Microfluidic Diaphragm Pressure Sensor for Health and Tactile Touch Monitoring

*Yuji Gao, Hiroki Ota, Ethan W. Schaler, Kevin Chen, Allan Zhao, Wei Gao, Hossain M. Fahad, Yonggang Leng, Anzong Zheng, Furui Xiong, Chuchu Zhang, Li-Chia Tai, Peida Zhao, Ronald S. Fearing, and Ali Javey**

Dr. Y. Gao, Dr. H. Ota, Dr. E. W. Schaler, Dr. K. Chen, A. Zhao, Dr. W. Gao, Dr. H. M. Fahad, C. Zhang, L. -C. Tai, P. Zhao, Prof. R. S. Fearing, Prof. A. Javey
Department of Electrical Engineering & Computer Sciences, University of California, Berkeley, CA 94720, USA
E-mail: ajavey@eecs.berkeley.edu
Dr. Y. Gao, Dr. H. Ota, Dr. K. Chen, Dr. W. Gao, Dr. H. M. Fahad, Prof. A. Javey
Berkeley Sensor and Actuator Center, University of California, Berkeley, CA 94720, USA
Dr. Y. Gao, Dr. H. Ota, Dr. K. Chen, Dr. W. Gao, L. -C. Tai, P. Zhao, Prof. A. Javey
Materials Sciences Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA
Dr. Y. Gao, Prof. Y. Leng, Dr. F. Xiong
School of Mechanical Engineering, Tianjin University, Tianjin 300072, China
Prof. Y. Leng
Key Laboratory of Mechanism Theory and Equipment Design of Ministry of Education, Tianjin University, Tianjin 300072, China
A. Zheng
National Center for Computer Animation, Bournemouth University, Bournemouth, BH12 5BB, UK

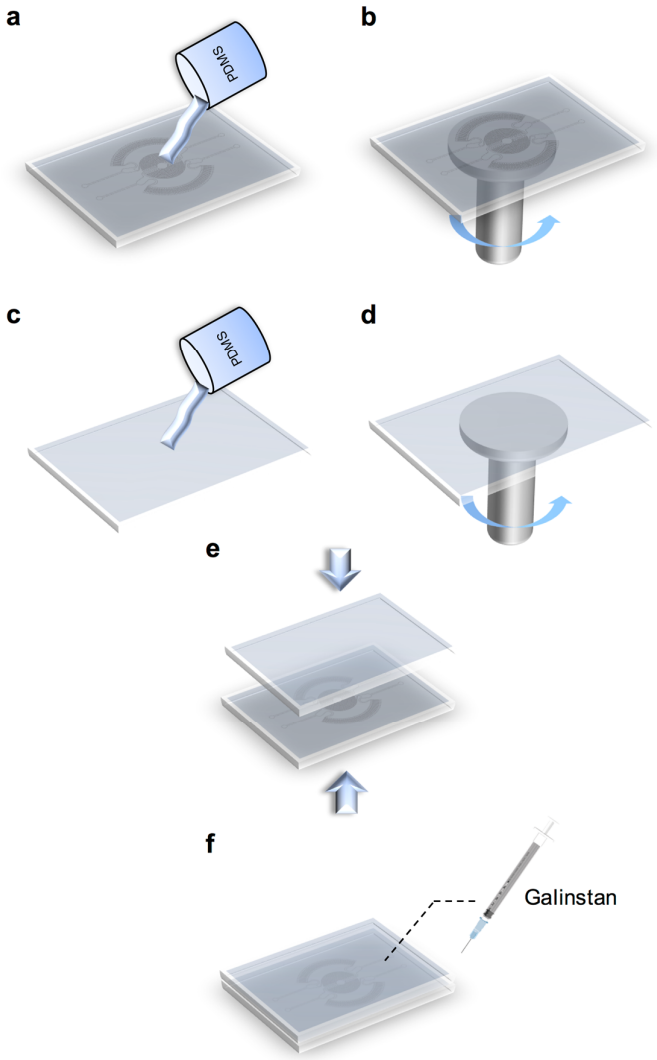


Figure S1. Fabrication process of microfluidic tactile diaphragm pressure sensor: a) Liquid PDMS is poured onto the SU-8 mold and b) spin coated at 500 rpm to form a 170 µm thick layer. c) Liquid PDMS is poured onto a bare glass slide and d) also spin coated at 500 rpm. e) After curing, the patterned PDMS membrane and the plain PDMS membrane are bonded together. f) Galinstan is injected into the microchannels.

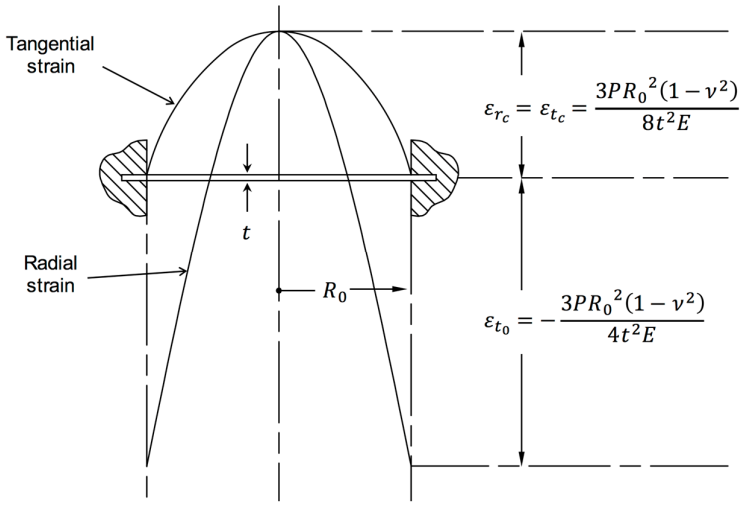


Figure S2. Strain distribution in a rigidly clamped diaphragm under uniform pressure.

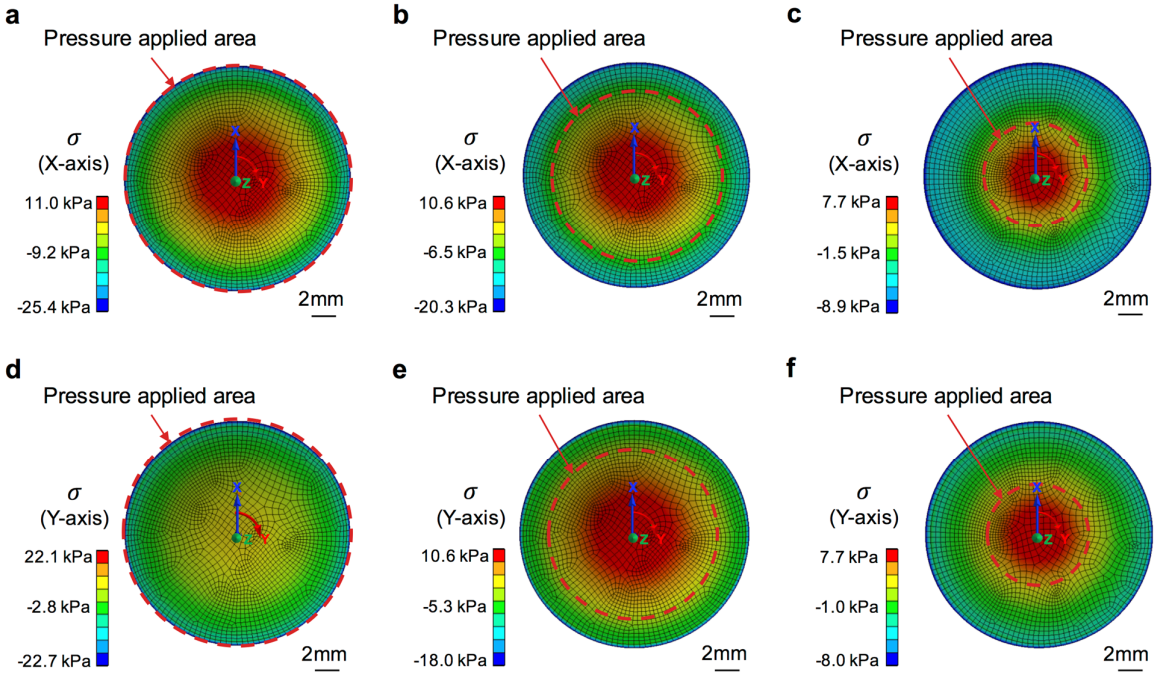


Figure S3. Simulations of the normal stress (σ) of 1 kPa pressure applied across different diameters of a 20 mm diameter sensor. a) Diameter of 20 mm (along X-axis), b) Diameter of 15 mm (along X-axis), c) Diameter of 9 mm (along X-axis), d) Diameter of 20 mm (along Y-axis), e) Diameter of 15 mm (along Y-axis), and f) Diameter of 9 mm (along Y-axis).

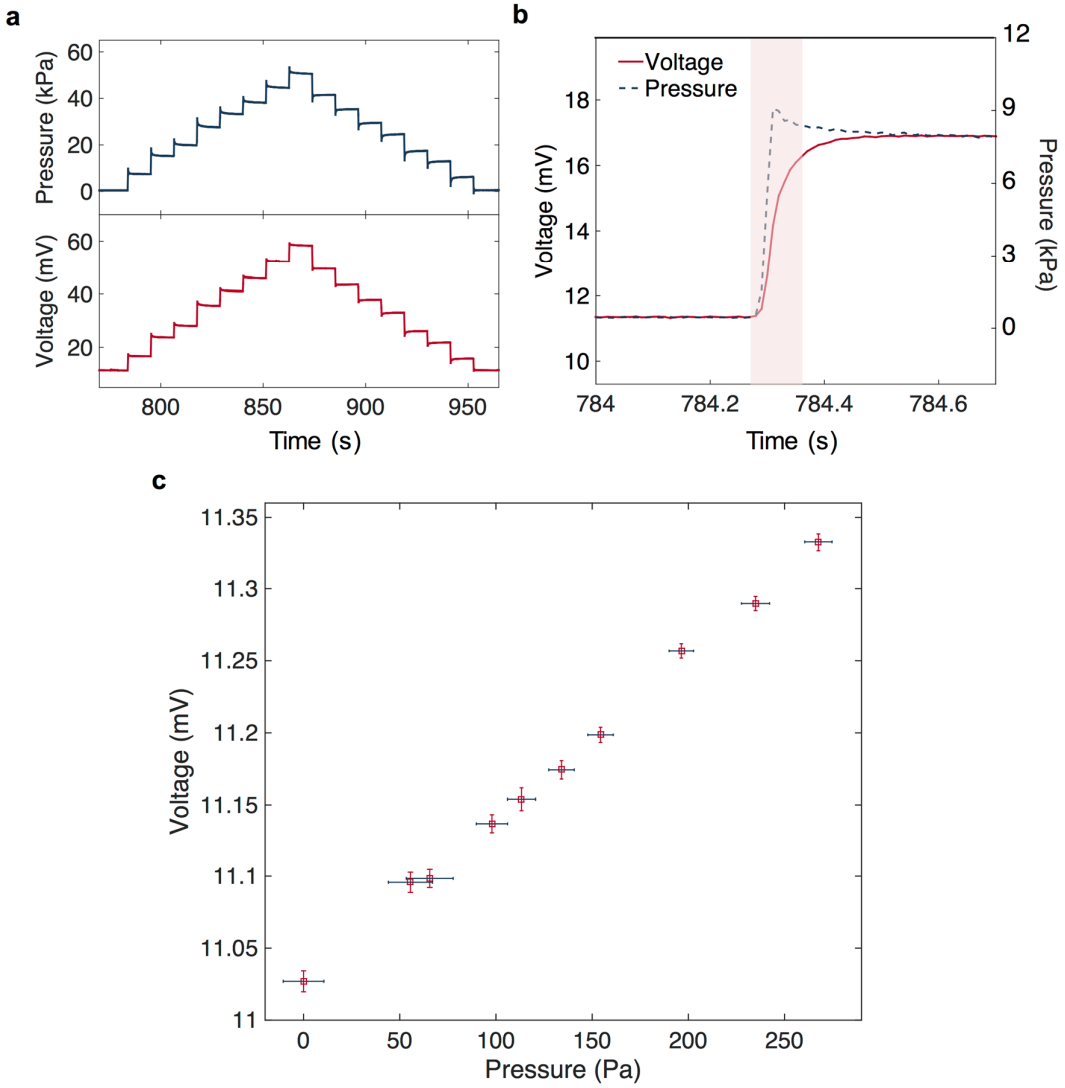


Figure S4. Extended characterization of the diaphragm pressure sensor. a) Real-time monitoring of the output voltage change in Cycle 5 of Fig. 2b, b) Response time of 90 ms derived from one step, and c) Sensor response indicating a detection limit of approximately 98 Pa and resolution of less than 50 Pa.

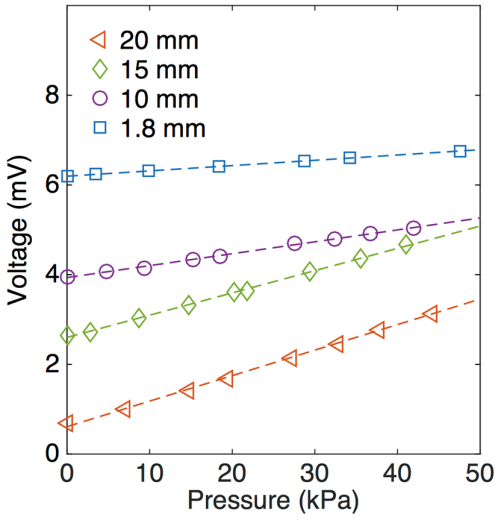
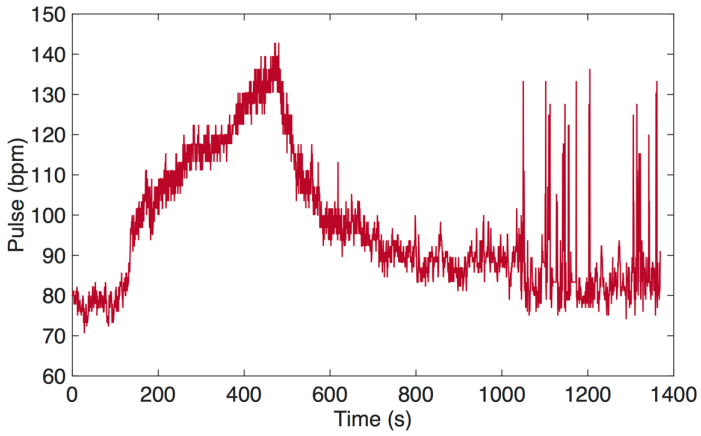


Figure S5. Effect of sensor size on sensitivity. Output voltage vs pressure for sensors with 1.8, 10, 15, and 20 mm diameter with 30 mV input voltage.



FigureS6. Dynamic pulse measurement in beat per min (bpm) before smoothing process.

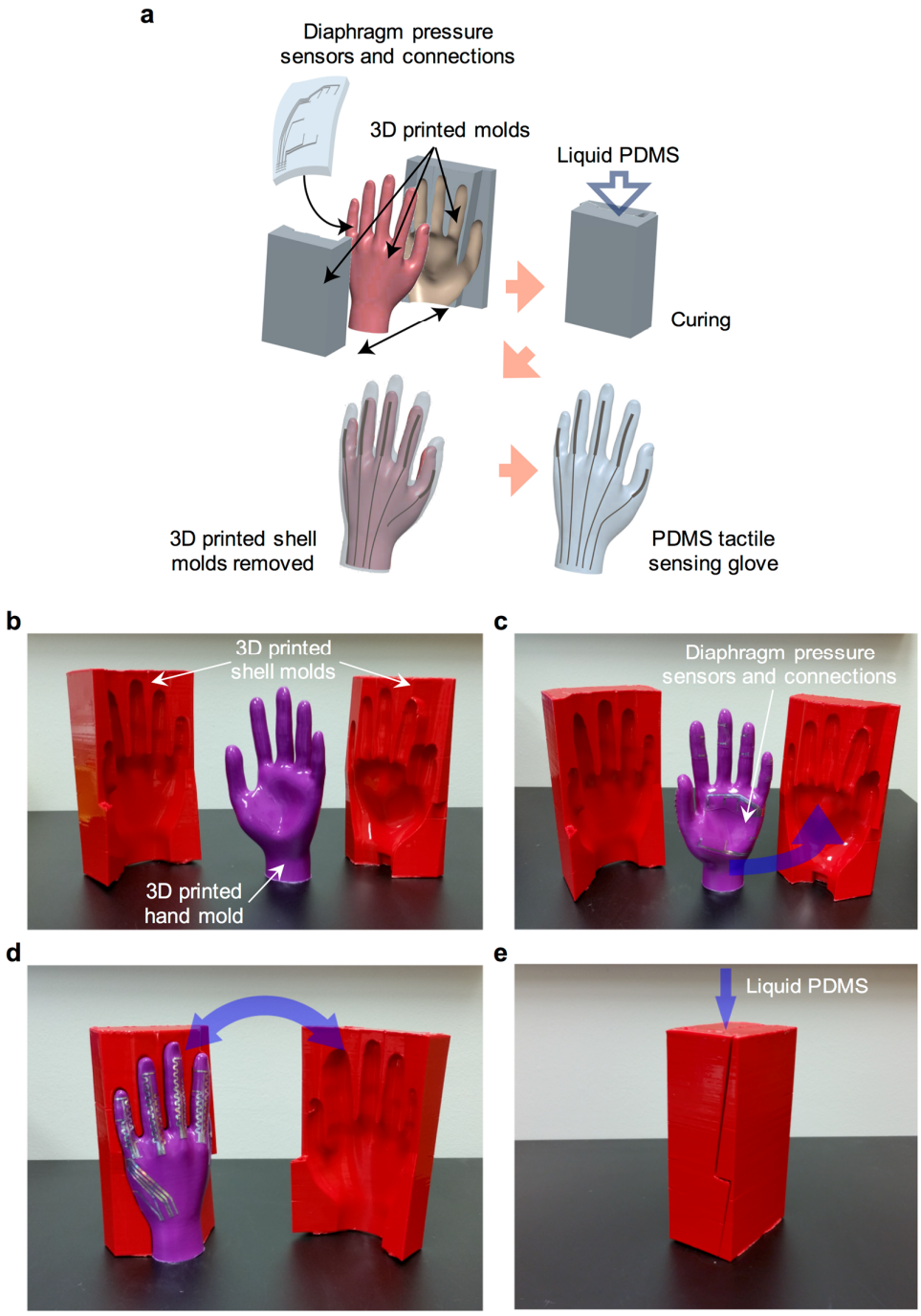


Figure S7. Fabrication process of a PDMS tactile sensing glove: a) Schematic of the glove fabrication process, b) Customized 3D printed ABS hand mold and shell molds after smoothing treatment (acetone evaporation), c) Six PDMS membranes fabricated separately beforehand with embedded sensors and connections placed between the hand mold and shell molds. d) Hand molds are sealed by the shell molds and e) Liquid PDMS is poured into the molds and cured.

Table S1. Comparison of flexible pressure sensors based on embedded conductive liquid microchannels.

Type of sensor	Functional material	Microchannel dimension	Sensitivity	Linearity	Limit of detection	Response time	Temperature reliability	Ref.
Capacitance	RTIL	100 µm × 15 µm [a)]	~0.23 nF/kPa	-	-	-	-	16
Capacitance	Galinstan	125 µm thick	~2.25×10^{-4} kPa^{-1} [b)]	R^2=0.982	-	-	-	22
Resistance	RTIL	150 µm × 70 µm [a)]	~2.62×10^{-4} kPa^{-1} [b)]	R^2>0.99	-	-	up to 100 °C	17
Resistance	EGaIn	100 µm × 80 µm [a)]	0.05 kPa^{-1}	R^2=0.993	4 kPa	-	15-45 °C	18
Resistance	EGaIn	100 µm × 80 µm [a)]	(2-20)×10^{-3} kPa^{-1}	-	2 kPa	-	19-45 °C	19
Resistance	EGaIn	200 µm × 300 µm [a)]	~0.18 kPa^{-1} [b)]	-	15 kPa	-	-	20
Resistance	EGaIn	200 µm × 200 µm [a)] 300 µm × 700 µm [a)]	0.79-16.18 mV/kPa [b)]	-	-	-	25-70 °C	21
Resistance	EGaIn	0.25 mm^2 area	~0.075 kPa^{-1} [b)]	-	-	-	-	27
Resistance	Galinstan	20 µm × 100 µm [a)]	~5.27 mV/kPa [b)]	R^2>0.98	-	< 0.5 s	-	23
Resistance	Galinstan	70 µm × 70 µm [a)]	0.0835 kPa^{-1} [c)] 0.0834 kPa^{-1} [d)]	R^2=0.99867 [e)] R^2=0.99918 [f)]	98 Pa	90 ms	20-50 °C	This work

[a)] Width × height; [b)] Approximated from paper data; [c)] Loading scenario; [d)] Unloading scenario; [e)] Loading scenario; [f)] Unloading scenario.

File S1. Matlab code for smoothing pulse measurements.

Main code

```
clear; clc; clf;

% read a comma separated value file.
% exclude the first two columns.
m = csvread('filename',0,2);

samples = size(m,1);    % sample size.
time = m(1:samples,1);  % sampled times.
data = m(1:samples,10); % sampled data.

% show sampled data.
figure(1);
plot(time, data);
xlabel('Time (s)','fontsize',16);ylabel('Voltage (mV)','fontsize',16);

%-------------------------------------------------------------
% In the program, the period representing heart beat durations
% is changing during time, so a dynamical method to accurately
% predict following period is required.
%-------------------------------------------------------------

% initial period.
% the period is a guess of possible number of samples to
% find a maximum representing a heart beat.
period = 70;
halfPeriod = ceil(period*0.5); % search expand.

% find the first four peaks to predict following periods.
peak = 1; % suppose initial peak position here.
max = data(1);
for i=1:1.5*period
   if max < data(i)
      max = data(i);
      peak = i;
   end
end

% first peak.
peak_pos = find_peak( data(1:ceil(1.5*period)) );
peaks = [peak_pos];

% second peak.
peak_pos = find_peak( data(peak_pos+ceil(period/2) : peak_pos+ceil(1.5*period)) ) +
```

```
peak_pos+ceil(period/2)-1;
periods = [peak_pos-peaks(end)];
peaks = [peaks, peak_pos];

% third peak.
peak_pos = find_peak( data(peak_pos+ceil(period/2) : peak_pos+ceil(1.5*period)) ) +
peak_pos+ceil(period/2)-1;
periods = [periods,peak_pos-peaks(end)];
peaks = [peaks, peak_pos];

% fourth peak.
peak_pos = find_peak( data(peak_pos+ceil(period/2) : peak_pos+ceil(1.5*period)) ) +
peak_pos+ceil(period/2)-1;
periods = [periods,peak_pos-peaks(end)];
peaks = [peaks, peak_pos];

% dynamically update period
done = 0;
while i<= samples

  % estimate next pMark
  nextPeriod = ceil((periods(end)+periods(end-1)+periods(end-2))/3);
  % search expansion.
  halfPeriod = ceil(nextPeriod*0.5);

  % search range between srart position and end position.
  start_pos = peak_pos+ceil(nextPeriod/2);
  end_pos = peak_pos+ceil(1.5*nextPeriod);

  if end_pos > samples
     end_pos = samples;
     done = 1;
  end

  peak_pos = find_peak( data( start_pos: end_pos ) ) + peak_pos+ceil(nextPeriod/2)-1;

  % if the computed period deviates from former peridos too much,
  % abandon it, use a position between start and end position to replace it.
  % 0.3 is the control threshold.
  if abs((peak_pos-peaks(end) - periods))/periods > 0.3
     peak_pos = ceil((start_pos+end_pos) / 2);
  end

  periods = [periods,peak_pos-peaks(end)];
  peaks = [peaks, peak_pos];
```

```
  if done == 1
     break;
  end
end

% show found peakes.
figure(2);
plot(time, data, '-', time(peaks), data(peaks),'o');
xlabel('Time (s)','fontsize',16);ylabel('Voltage (mV)','fontsize',16);

% convert periods to beat rates.
beats = zeros(1, size(peaks,2)-1);

for i=1:size(periods,2)
  beats(i) = 6000 / periods(i);% peaks(i+1)-peaks(i);
end

% show beat rates.
ntime = peaks*0.01;
figure(3); plot(ntime(1:end-1), beats);
xlabel('Time (s)','fontsize',16);ylabel('Pulse (bpm)','fontsize',16);
axis([0,1400,60,150]);

% use a lowess smooth scheme to filter the data.
% the smooth size is chosen as 30.
smooth_data = smooth(beats,30,'lowess');

% show smoothed heart rates.
figure(4);
plot(ntime(1:end-1), smooth_data);
xlabel('Time (s)','fontsize',16);ylabel('Pulse (bpm)','fontsize',16);
axis([0,1400,60,150]);
```

Subcode

```
% find peak position inside an input sequence.
function pos = find_peak( x )

max = x(1);
pos = 1;

for i=1:size(x)
  if max < x(i)
     max = x(i);
     pos = i;
  end
end
```