

Matlab Code

```
1 %% Pre-process
2 close all; % close all figures
3 clear all; % clear all workspace variables
4 clc; % clear the command line
5 fclose('all'); % close all open files
6 delete(instrfindall); % reset com port
7
8 %% Constatns
9 BAUDRATE = 57600; % Need to configure Xbee parts BD i.e.
    Interface Data Rate
10 INPUTBUFFER = 51200; %Unit in bytes. Definition:A
    location that holds all
11 % incoming information before it continues to the CPU for
    processing;
12 % Buffers used to store information before it processed
13
14
15 %% Initialize the stripchart
16 figure_acceleration = figure('Name','Acceleration');
17 axes_accelerationx = subplot(3,1,1); % Axes Object
18 xlabel(axes_accelerationx,'Time/ second')
19 ylabel(axes_accelerationx,'Acceleration/ m/s^2')
20
21 axes_accelerationy = subplot(3,1,2);
22 xlabel(axes_accelerationy,'Time/ second')
23 ylabel(axes_accelerationy,'Acceleration/ m/s^2')
24
25 axes_accelerationz = subplot(3,1,3);
26 xlabel(axes_accelerationz,'Time/ second')
27 ylabel(axes_accelerationz,'Accelerationz/ m/s^2')
28
29 h_ax = animatedline(axes_accelerationx,'MaximumNumPoints'
    ,Inf,'Marker','o','Color','red'); % animated line
    object
30 h_ay = animatedline(axes_accelerationy,'MaximumNumPoints'
    ,Inf,'Marker','+','Color','blue');
31 h_az = animatedline(axes_accelerationz,'MaximumNumPoints'
    ,Inf,'Marker','*','Color','green');
32
33 figure_gyro = figure('Name','Gyro');
34 axes_gyrox = subplot(3,1,1);
35 xlabel(axes_gyrox,'Time/ second')
36 ylabel(axes_gyrox,'Gyrox/ rad/s')
```

```

37
38 axes_gyro_y = subplot(3,1,2);
39 xlabel(axes_gyro_y, 'Time/ second')
40 ylabel(axes_gyro_y, 'Gyro/ rad/s')
41
42 axes_gyro_z = subplot(3,1,3);
43 xlabel(axes_gyro_z, 'Time/ second')
44 ylabel(axes_gyro_z, 'Gyro/ rad/s')
45
46 h_gx = animatedline(axes_gyro_x, 'MaximumNumPoints', Inf, '
    Marker', 'o', 'Color', 'red');
47 h_gy = animatedline(axes_gyro_y, 'MaximumNumPoints', Inf, '
    Marker', '+', 'Color', 'blue');
48 h_gz = animatedline(axes_gyro_z, 'MaximumNumPoints', Inf, '
    Marker', '*', 'Color', 'green');
49
50 figure_orientation = figure('Name', 'Orientation');
51 axes_roll = subplot(3,1,1);
52 xlabel(axes_roll, 'Time/ second')
53 ylabel(axes_roll, 'Roll/ deg')
54
55 axes_pitch = subplot(3,1,2);
56 xlabel(axes_pitch, 'Time/ second')
57 ylabel(axes_pitch, 'Pitch/ deg')
58
59 axes_heading = subplot(3,1,3);
60 xlabel(axes_heading, 'Time/ second')
61 ylabel(axes_heading, 'Heading/ deg')
62
63 h_roll = animatedline(axes_roll, 'MaximumNumPoints', Inf, '
    Marker', 'o', 'Color', 'red');
64 h_pitch = animatedline(axes_pitch, 'MaximumNumPoints', Inf, '
    Marker', '+', 'Color', 'blue');
65 h_heading = animatedline(axes_heading, 'MaximumNumPoints',
    Inf, 'Marker', '*', 'Color', 'green');
66
67 % Index
68 i = 1;
69 j = 1;
70 k = 1;
71
72 % Record the data
73 % 10000 Row data
74 record_acceleration = zeros(10000,4);
75 record_gyro = zeros(10000,4);
76 record_orientation = zeros(10000,4);

```

```

77
78 %% Create a serial object
79 board = serial('COM4','BaudRate',BAUDRATE,'DataBits',8);
80 % Related with the used COM, can be different
81 % COM4 for wireless, COM3 for wire
82 % BAUDRATE unit: bits per second, e.g. 9600 means 9600
    bits per second,
83 % i.e. 1200 bytes per second i.e. 1.2 KB/s or 1.17KB/s
84
85
86 % Set serial port buffer
87 set(board,'InputBufferSize',INPUTBUFFER);
88 % InputBufferSize: total number of bytes that can be
    stored in the input
89 % buffer during a read operation. The read operation is
    terminated if the
90 % amount of data stored in the input buffer equals the
    InputBufferSize
91
92 fopen(board);
93 while(1)
94     a = fgetl(board);% a:character vector
95     b = textscan(a,'%u32 %s %f %f %f %u32','Delimiter',' ',
        ' ');
96
97     if strcmp(char(b{2}),'Acceleration')
98
99         time = double(b{1})/1000; % in second
100         ax = b{3};
101         addpoints(h_ax,time,ax);
102         drawnow limitrate
103
104         ay = b{4};
105         addpoints(h_ay,time,ay);
106         drawnow limitrate
107
108         az = b{5};
109         addpoints(h_az,time,az);
110         drawnow limitrate
111
112         record_acceleration(i,:)=[time ax ay az];
113         i=i+1;
114
115     elseif strcmp(char(b{2}),'Gyro')
116         time = double(b{1})/1000; % in second
117         gx = b{3};

```

```

118         addpoints(h_gx,time,gx);
119         drawnow limitrate
120
121     gy = b{4};
122         addpoints(h_gy,time,gy);
123         drawnow limitrate
124     gz = b{5};
125         addpoints(h_gz,time,gz);
126         drawnow limitrate
127
128     record_gyro(j,:)= [time gx gy gz];
129     j=j+1;
130
131     elseif strcmp(char(b{2}), 'Orientation')
132         time = double(b{1})/1000; % in second
133         roll = b{3};
134         addpoints(h_roll,time,roll);
135         drawnow limitrate
136
137         pitch = b{4};
138         addpoints(h_pitch,time,pitch);
139         drawnow limitrate
140
141         heading = b{5};
142         addpoints(h_heading,time,heading);
143         drawnow limitrate
144
145         record_orientation(k,:) = [time roll pitch
146                                     heading];
146         k = k+1;
147     end
148
149 end

```