```
// "A Data Processor in 3D Calligraphy"
// This is a sketch of Processing to get 3D track of "A Writing Brush".
// A CSV file "rawData.csv" containig panel data from Arduino should be in the same folder of this sketch.
// This sketch makes an output CSV file "syodo.csv" in the folder above.
// Copyright (C) 2014 ArduinoDeXXX All Rights Reserved.

PrintWriter output;  //1
int LENGTH;  //2
String [][] csv;  //3
int k = 0;  //4
int h = 0;  //5
int m = 0;
float t, fwd_t;
float rx, ry, rz;
float bs_rx, bs_ry, bs_rz;
float Rx, Ry, Rz;  //10
float ax_0, ay_0, az_0;
float ax, ay, az;
float nax, nay, naz;
float nnax, nnay, nnaz;
float vx, vy, vz;  //15
float X, Y, Z;
float ox, oy, oz;
float xx, xy, xz;
float yx, yy, yz;
float zx, zy, zz;  //20
double g_0, ga, g;
int shortStop = 0;
byte countShrtStp = 0;
int maxStop = 280;
int miniStop = 50;
int [] rec_sStopS = new int[128];
int [] rec_sStopF = new int[128];
int startObs = 0;
float ax_w, ay_w, az_w;
float ax_1, ay_1, az_1;  //30
float gx_w, gy_w, gz_w;
float gx, gy, gz;
int shortPush = 65;  //33 <--- The max number of observations in clicking a push switch at start or end of drawing
byte countWrite = 0;
int [] rec_on = new int[128];
int [] rec_off = new int[128];
int [] rec_shrt = new int[128];
int [] rec_wrtS = new int[128];
int [] rec_wrtF = new int[128];
int sw = 0;  //40
int iiObs = 50;
int iObs = 450;
float cnvtr = 875 * 1.02;  //43 <--- conversion factor to get angular velocity from digital output
float cF =  0.1;
float lcSc = 4;
int winW = 1200;
int winH = winW * 10 / 16;
float ini_X = 0.5;
float ini_Y = 0.5;
float ini_Z = 0.5;  //50

void setup(){  //51
  size(winW, winH, P3D);
int csvWidth = 0;
String lines[] = loadStrings("rawData.csv");
for (int i=0; i < lines.length; i++) {
  String [] chars = split(lines[i],',');
  if (chars.length > csvWidth){ csvWidth = chars.length; }
}
csv = new String [lines.length][csvWidth];
LENGTH =lines.length;  //60
println(LENGTH);
for (int i=0; i < lines.length; i++) {
  String [] temp = new String [lines.length];
  temp= split(lines[i], ',');
  for (int j=0; j < temp.length; j++){ csv[i][j]=temp[j]; }
}
for (int i = iiObs ; i < iiObs + iObs ; i++) {
  ax_0 = ax_0 - Float.parseFloat(csv[i][4]);
  az_0 = az_0 + Float.parseFloat(csv[i][5]);
  ay_0 = ay_0 - Float.parseFloat(csv[i][6]);  //70
  bs_rx = bs_rx - Float.parseFloat(csv[i][1]);
  bs_rz = bs_rz + Float.parseFloat(csv[i][2]);
  bs_ry = bs_ry + Float.parseFloat(csv[i][3]);
}
ax_0 = ax_0 / iObs;
ay_0 = ay_0 / iObs;
az_0 = az_0 / iObs;
gx_w = ax_0;
```

```
  gy_w = ay_0;
  gz_w = az_0;  //80
  g_0 = sqrt( (ax_0 * ax_0) + (ay_0 * ay_0) + (az_0 * az_0) );
  bs_rx = bs_rx / iObs;
  bs_ry = bs_ry / iObs;
  bs_rz = bs_rz / iObs;
  k = iiObs + iObs;
  for ( int i=0; i<128; i++ ) {
    rec_on[i] = 0;
    rec_off[i] = 0;
    rec_shrt[i] = 0;
    rec_wrtS[i] = 0;  //90
    rec_wrtF[i] = 0;
    rec_sStopS [i] = 0;
    rec_sStopF [i] = 0;
  }
  output = createWriter("syodo.csv");
  frameRate(120);
  scanOnOff();
  scanStop();
}  //99

void draw(){  //100
  background(255);
  translate( winW * ini_X,   winH * ini_Y,   -winW * ini_Z );
  vx = 0;    vy = 0;    vz = 0;
  X = 0;     Y = 0;     Z = 0;
  ax_1 = 0;  ay_1 = 0;  az_1 = 0;
  startObs = 0;
  k++;
  if ( k == rec_wrtF[ countWrite ] + 1 ){ finProcess(); }
  if ( k > rec_wrtF[h] ) {
    h++;  //110
    k = rec_wrtS[h];
  }
  for ( m = 0 ; m <= countShrtStp ; m++ ) {
    if ( rec_sStopF[m] < rec_wrtS[h] ) {
      startObs = rec_sStopS[m];
      shortStop = rec_sStopF[m] - rec_sStopS[m] + 1;
    }
  }
  for ( int i = startObs ; i <= k ; i++ ) {
    if ( i == rec_wrtS[h] ) {    //120
      vx = 0;    vy = 0;    vz = 0;
      X = 0;     Y = 0;     Z = 0;
    }
    t = Float.parseFloat(csv[i][0]);
    fwd_t = Float.parseFloat(csv[i+1][0]);
    t = fwd_t - t;
    rx = - Float.parseFloat(csv[i][1]);
    rz =   Float.parseFloat(csv[i][2]);
    ry =   Float.parseFloat(csv[i][3]);
    rx = rx - bs_rx;  //130
    ry = ry - bs_ry;
    rz = rz - bs_rz;
    ax = - Float.parseFloat(csv[i][4]);
    az =   Float.parseFloat(csv[i][5]);
    ay = - Float.parseFloat(csv[i][6]);
    ga = sqrt( (ax * ax) + (ay * ay) + (az * az) );
    rx = ( rx / 1000 ) * (cnvtr / 100);
    ry = ( ry / 1000 ) * (cnvtr / 100);
    rz = ( rz / 1000 ) * (cnvtr / 100);
    Rx = rx * (t / 1000);  //140
    Ry = ry * (t / 1000);
    Rz = rz * (t / 1000);
    rotateX( radians( Rx/1000 ) );
    rotateY( radians( Ry/1000 ) );
    rotateZ( radians( Rz/1000 ) );
    ox = modelX(0,0,0);
    oy = modelY(0,0,0);
    oz = modelZ(0,0,0);
    xx = modelX(1,0,0) - ox;
    xy = modelY(1,0,0) - oy;
    xz = modelZ(1,0,0) - oz;  //150
    yx = modelX(0,1,0) - ox;
    yy = modelY(0,1,0) - oy;
    yz = modelZ(0,1,0) - oz;
    zx = modelX(0,0,1) - ox;
    zy = modelY(0,0,1) - oy;
    zz = modelZ(0,0,1) - oz;
    gx = ( gx_w * xx ) + ( gy_w * xy ) + ( gz_w * xz );
    gy = ( gx_w * yx ) + ( gy_w * yy ) + ( gz_w * yz );
    gz = ( gx_w * zx ) + ( gy_w * zy ) + ( gz_w * zz );
    g = sqrt( (gx * gx) + (gy * gy) + (gz * gz) );  //160
    if ( i > rec_wrtS[h] ) {
```

```
  nax = ( ( ax - gx ) / 16 ) * (t / 1000);
  nay = ( ( ay - gy ) / 16 ) * (t / 1000);
  naz = ( ( az - gz ) / 16 ) * (t / 1000);
  nnax = nax - vx * cF;
  nnay = nay - vy * cF;
  nnaz = naz - vz * cF;
  vx = vx + ( nnax / 1000 );
  vy = vy + ( nnay / 1000 );
  vz = vz + ( nnaz / 1000 );  //170
  X = vx * (t / 1000);
  Y = vy * (t / 1000);
  Z = vz * (t / 1000);
  translate( (X/1000)*lcSc, (Y/1000)*lcSc, (Z/1000)*lcSc );
}
sw = (int)Float.parseFloat(csv[i][7]);
if ( i == rec_wrtF[h] ) { sw = 0; }
if ( i <= rec_sStopF[m] ) {
  ax_1 = ax_1 + ax;
  ay_1 = ay_1 + ay;  //180
  az_1 = az_1 + az;
}
if ( i == rec_sStopF[m] ) {
  ax_1 = ax_1 / shortStop;
  ay_1 = ay_1 / shortStop;
  az_1 = az_1 / shortStop;
  ax_w = modelX(ax_1,  ay_1,  az_1);
  ay_w = modelY(ax_1,  ay_1,  az_1);
  az_w = modelZ(ax_1,  ay_1,  az_1);
  gx_w = ( ( 10 * gx_w ) + ( shortStop * ax_1 ) ) / ( 10 + shortStop );  //190
  gy_w = ( ( 10 * gy_w ) + ( shortStop * ay_1 ) ) / ( 10 + shortStop );
  gz_w = ( ( 10 * gz_w ) + ( shortStop * az_1 ) ) / ( 10 + shortStop );
  ax_1 = 0;  ay_1 = 0;  az_1 = 0;
 }
}
fill(255,240,200);
stroke(0,0,0);
box(40, 10, 90);
strokeWeight(2.5);
line(0,0,0,  70,0,0);  //200
stroke(255,0,0);
line(0,0,0,  0,55 ,0);
stroke(0,0,255);
line(0,0,0,  0,0,95);
strokeWeight(1);
stroke(0,255,255);
line(0,0,0,   gx/100,  gy/100,  gz/100);
line(0,0,0,  -gx/100, -gy/100, -gz/100);
textAlign(CENTER);
fill(#EB0FF0);  //210
textSize(30);
text( (int)fwd_t/100000,   gx/100,  gy/100 -5,  gz/100);
textSize(20);
text( "/10 sec",   gx/100 + 60,  gy/100 -5,  gz/100);
textSize(30);
text( (int)(vx*9.8/10),   -gx/100 - 70,  -gy/100 + 55,  -gz/100);
text( (int)(vy*9.8/10),   -gx/100,       -gy/100 + 55,  -gz/100);
text( (int)(vz*9.8/10),   -gx/100 + 70,  -gy/100 + 55,  -gz/100);
textSize(20);
text( "cm/s",   -gx/100 + 130,  -gy/100 + 55,  -gz/100);
 textSize(30);  //220
 text( "X     Y     Z",   -gx/100,  -gy/100 + 25,  -gz/100);
 if ( vx > 0 ) { text( "→",   -gx/100 -70,  -gy/100 + 85,  -gz/100);
   } else {      text( "←",   -gx/100 -70,  -gy/100 + 85,  -gz/100); }
 if ( vy > 0 ) { text( "↓",   -gx/100,      -gy/100 + 85,  -gz/100);
   } else {      text( "↑",   -gx/100,      -gy/100 + 85,  -gz/100); }
 if ( vz > 0 ) { text( "○",   -gx/100 +70,  -gy/100 + 85,  -gz/100);
   } else {      text( "×",   -gx/100 +70,  -gy/100 + 85,  -gz/100); }
 output.print( k );   output.print(", ");  //228
 output.print( ox );  output.print(", ");  output.print( oy );  output.print(", ");  output.print( oz );  output.print(", ");
 output.print( xx );  output.print(", ");  output.print( xy );  output.print(", ");  output.print( xz );  output.print(", ");
 output.print( yx );  output.print(", ");  output.print( yy );  output.print(", ");  output.print( yz );  output.print(", ");
 output.print( zx );  output.print(", ");  output.print( zy );  output.print(", ");  output.print( zz );  output.print(", ");
 output.print( sw );    output.print(", ");  //233
 if ( k == rec_wrtF[h] ) { output.print(1); } else { output.print(0); }
 output.print(", ");  output.println( h );
}  //236

void scanOnOff() {  //237
  int onOff = 0;
  byte countOn = 0;
  byte countShtPsh = 0;  //240
  byte countEnd = 0;
  countWrite = 0;
  for ( int i = iiObs + iObs + 1; i < LENGTH ; i++ ) {
    onOff = (int)Float.parseFloat(csv[i][7]) -  (int)Float.parseFloat(csv[i-1][7]);
```

```
if ( onOff == 1 ) {
  countOn++;
  rec_on[ countOn ] = i;
  print("ON ");
  println(countOn);
  println(rec_on[ countOn ]);  //250
}
if ( onOff == -1 ) {
  rec_off[ countOn ] = i;
  print("OFF ");
  println(countOn);
  println(rec_off[ countOn ]);
  if ( rec_off[ countOn ] - rec_on[ countOn ] < shortPush ) {
    countShtPsh++;
    rec_shrt[ countShtPsh ] = i;
    if ( rec_shrt[ countShtPsh ] - rec_shrt[ countShtPsh - 1 ] < shortPush * 1.85 ) {  //260
      countWrite++;
      countEnd--;
      rec_wrtS[ countWrite ] = i;
      rec_wrtF[ countWrite - 1 ] = rec_on[ countOn - 2 ];
      print("START [ ");
      print(countWrite);
      print(" ] with obs.{ ");
      print(rec_wrtS[ countWrite ]);
      print(" }, latest END obs.{ ");
      print(rec_wrtF[ countWrite - 1 ]);  //270
      println(" }");
    } else {
      rec_wrtF[ countWrite ] = rec_on[ countOn ];
      countEnd++;
      if ( countEnd == countWrite ) {
        print("END [ ");
        print(countWrite);
        print(" ] with obs.{ ");
        print(rec_wrtF[ countWrite ]);  //280
        println(" }");
      } else {
        print("ShortPush [ ");
        print(countShtPsh);
        print(" ] with obs.{ ");
        print(rec_shrt[ countShtPsh ]);
        println(" }");
      }
    }
  }  //290
 }
 }
}  //293

void scanStop() {  //294
  shortStop = 0;
  countShrtStp = 0;
  for ( int i = iiObs + iObs + 1; i < LENGTH ; i++ ) {
    ax = - Float.parseFloat(csv[i][4]);
    az =   Float.parseFloat(csv[i][5]);
    ay = - Float.parseFloat(csv[i][6]);  //300
    ga = sqrt( (ax * ax) + (ay * ay) + (az * az) );
    if ( abs( (float)(ga - g_0) ) < maxStop ) {
      shortStop++;
    } else {
      if ( shortStop > miniStop ) {
        countShrtStp++;
        rec_sStopS [ countShrtStp ] = i - shortStop;
        rec_sStopF [ countShrtStp ] = i - 1;
      }
      shortStop = 0;  //310
    }
  }
}  //313

void finProcess() {  //314
  output.flush();
  output.close();  // Finishes the file
  exit();
}  //318
// Copyright (C) 2014 ArduinoDeXXX All Rights Reserved.
```