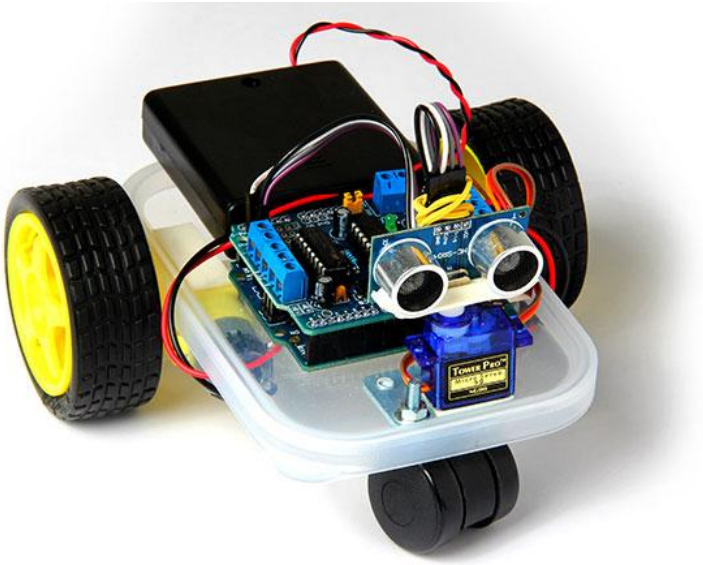
EP486 Microcontroller Applications

Topic 2

Processing



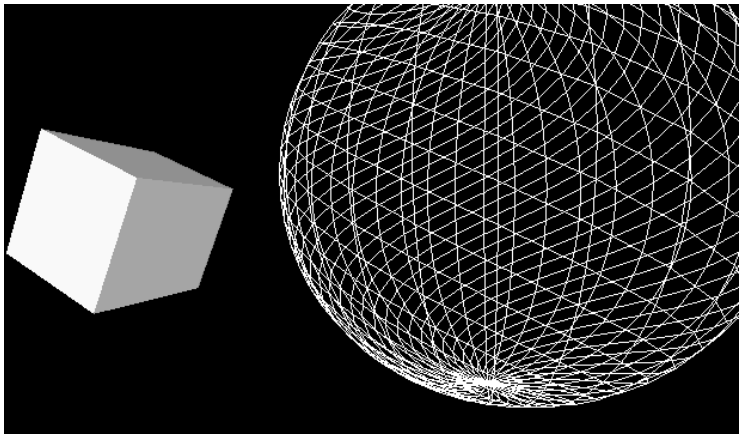
Department of Engineering Physics

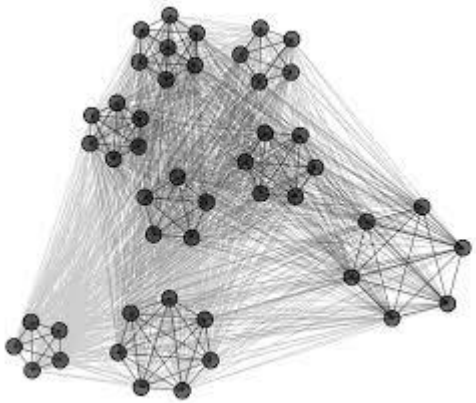
University of Gaziantep

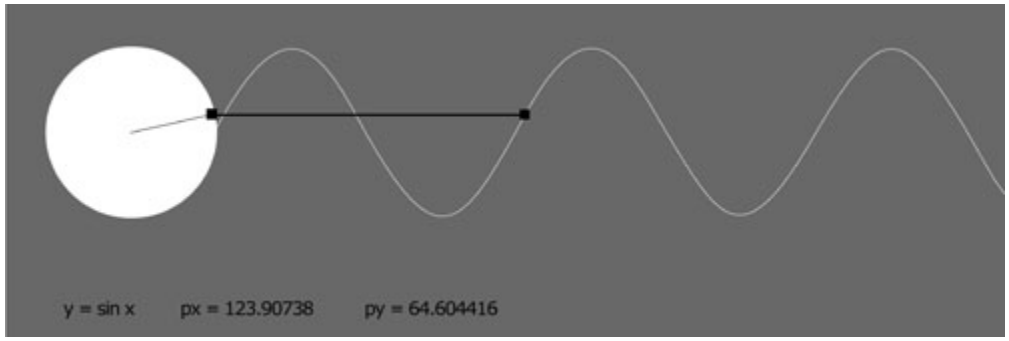
Sep 2013

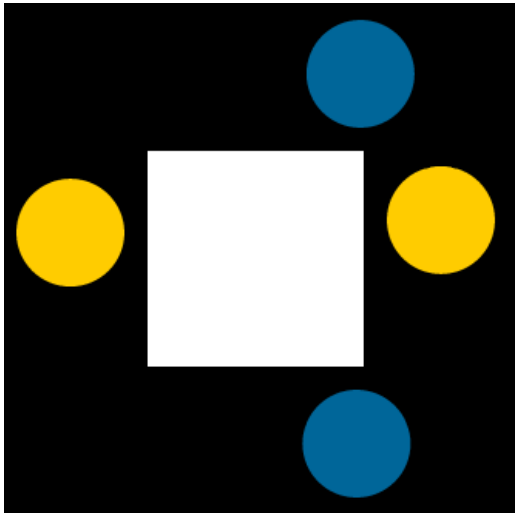
Processing is a programming language, development environment, and online community.

WebPage: http://processing.org/



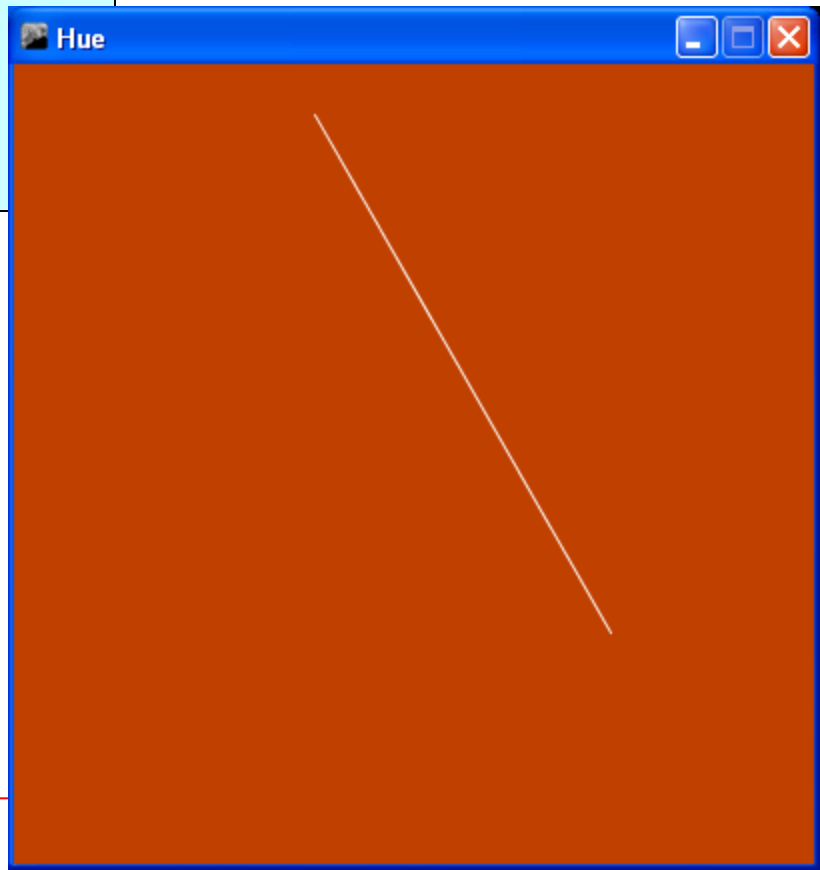






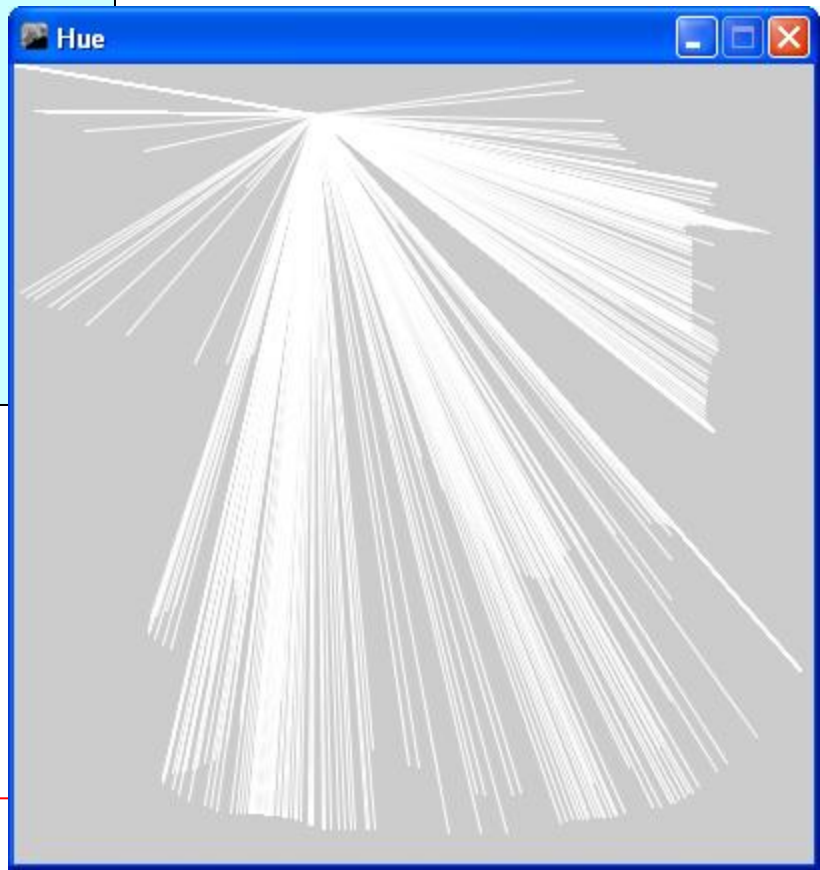
Example 1

```
void setup() {
   size(400, 400);
   stroke(255);
}

void draw() {
  background(192, 64, 0);
  line(150, 25, mouseX, mouseY);
}
```



Example 2

```
void setup() {
   size(400, 400);
   stroke(255);
}

void draw() {
  line(150, 25, mouseX, mouseY);
}

void mousePressed() {
   background(192, 64, 0);
}
```

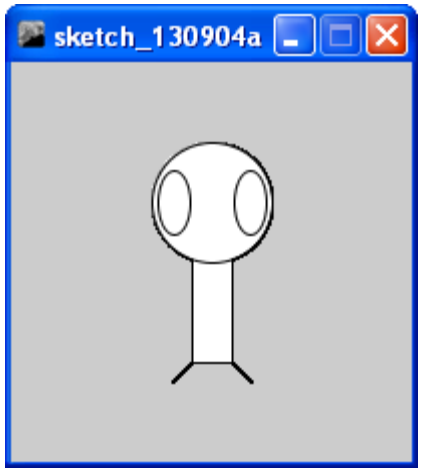


Example 3

```
void setup() {
  size(480, 120);
}
void draw()
{
   if (mousePressed) {
       fill(0);
   }
   else {
    fill(255);
  }
   ellipse(mouseX, mouseY, 80, 80);
}
```

Example 4

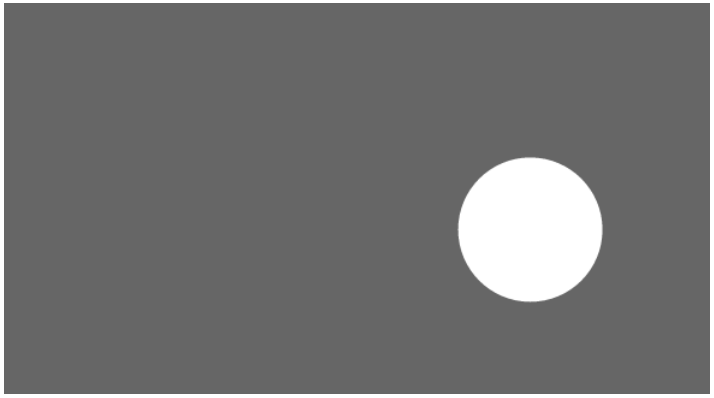
```
void setup() {
   size(200,200);
}
void draw()
{
  rectMode(CENTER);
  rect(100,100,20,100);
  ellipse(100,70,60,60);
  ellipse(81,70,16,32);
  ellipse(119,70,16,32);
  line(90,150,80,160);
  line(110,150,120,160);
}
```



Example 5

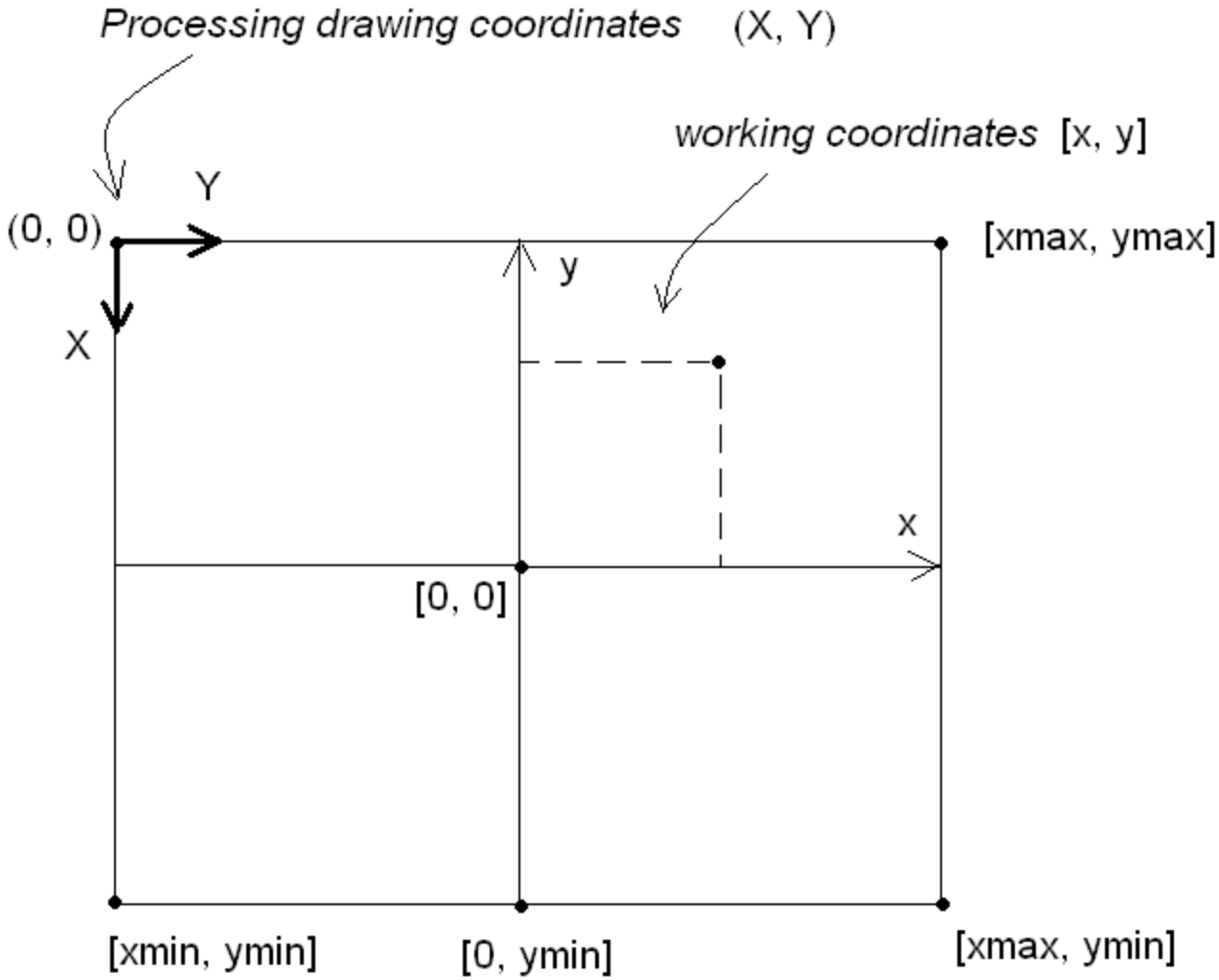
```
float a;

void setup() {
  size(640, 360);
  stroke(255);
  a = height/2;
}

void draw() {
  background(51);
  line(0, a, width, a);
  a = a - 0.5;
  if (a < 0) {
    a = height;
  }
}
```

Example 6



```
int rad = 60;          // Width of the shape
float xpos, ypos;      // Starting position of shape
float xspeed = 2.8;    // Speed of the shape
float yspeed = 2.2;    // Speed of the shape
int xdirection = 1;    // Left or Right
int ydirection = 1;    // Top to Bottom

void setup()
{
  size(640, 360);
  noStroke();
  frameRate(30);
  ellipseMode(RADIUS);
  xpos = width/2;
  ypos = height/2;
}

void draw()
{
  background(102);
  xpos = xpos + ( xspeed * xdirection );
  ypos = ypos + ( yspeed * ydirection );

  if (xpos > width-rad || xpos < rad) {
    xdirection *= -1;
  }
  if (ypos > height-rad || ypos < rad) {
    ydirection *= -1;
  }
  ellipse(xpos, ypos, rad, rad);
}
```

Coordinate Transformation



Transformation equations :

X = A*x + alpha
Y = B*y + beta

Transform Class

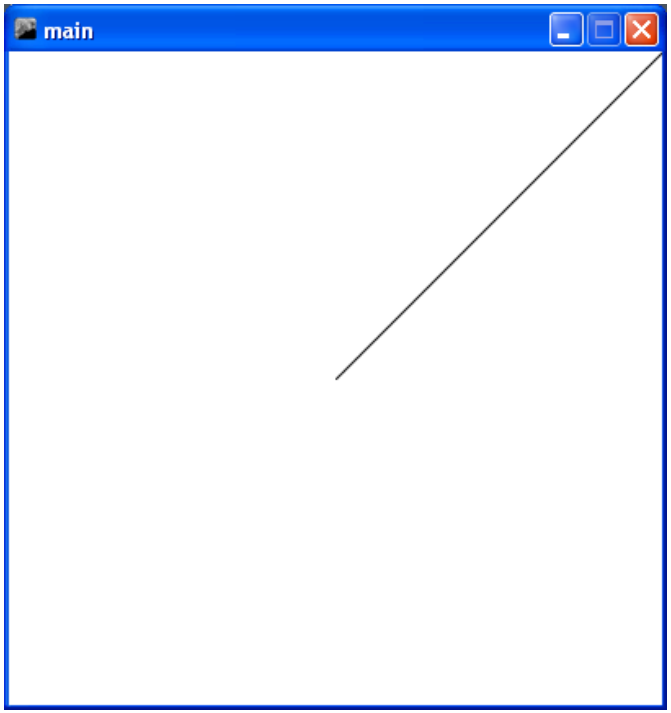
See also

ep486

home page

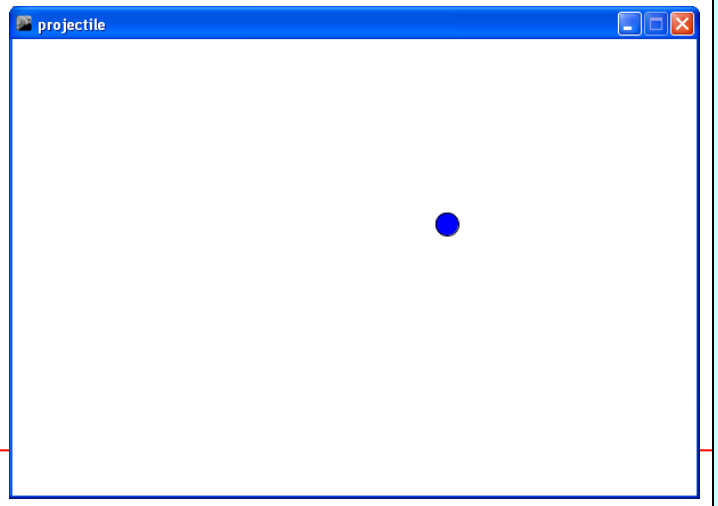
```
// Transform class ----------------------------------------
// Sep 2013
class Transform{
  float m_x, m_y, m_xmin,m_ymin, m_xmax, m_ymax;
  int   m_X, m_Y;
  Transform(float Xmin, float Xmax, float Ymin, float Ymax){
    m_xmin = Xmin;
    m_ymin = Ymin;
    m_xmax = Xmax;
    m_ymax = Ymax;
  }

  void transformX(float xx){
    int p = 0, q = 0;
    int r = p + width;
    int s = q + height;
    float a = (r-p)/(m_xmax-m_xmin);
    float alfa = p - a*m_xmin;
    m_X = int(a*xx + alfa);
  }

  void transformY(float yy){
    int p = 0, q = 0;
    int r = p + width;
    int s = q + height;
    float d = (s-q)/(m_ymin-m_ymax);
    float beta = q - d*m_ymax;
    m_Y = int(d*yy + beta);
  }

  int getX(float xx){
    transformX(xx);
    return m_X;
  }

  int getY(float yy){
    transformY(yy);
    return m_Y;
  }

  boolean outofRange(float xx, float yy){
    if(xx<m_xmin || xx>m_xmax || yy<m_ymin || yy>m_ymax) return true;
    return false;
  }
  boolean outofRange(){
    if(m_X>width || m_Y>height) return true;
    return false;
  }
}
// End of class
// ----------------------------------------------------------
```

Example 7




```
// Drawing a line
int X1,Y1, X2,Y2;
Transform ct;

void setup()
{
  size(400,400);
  ct = new Transform(-10,10, -10,10);
  stroke(0);
}

void draw()
{
  background(255);

  X1 = ct.getX(0.0);
  Y1 = ct.getY(0.0);
  X2 = ct.getX(10.0);
  Y2 = ct.getY(10.0);

  line(X1,Y1, X2,Y2);
}
```

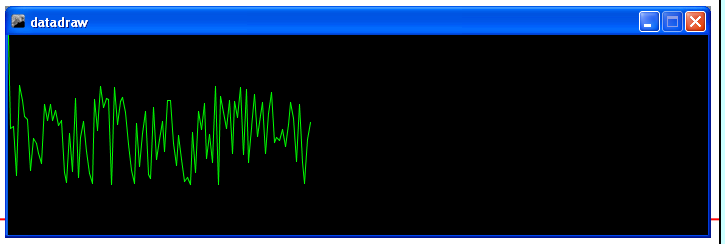
Example 8




```
// Projectile motion

float pi = 3.1415923;
float v0 = 80;  // initial velocity (m/s)
float theta = 40*pi/180; // initial angle
float g = 9.8; // gravitational acceleration
float t = 0.0; // time
float w = 20;  // width of the object
float x,y;
int X, Y;
Transform ct;

void setup() {
  size(600,400);
  ct = new Transform(0,800, 0,150);
  stroke(0);
  fill(0,0,255);
}

void draw(){
  background(255);
  x = v0*cos(theta)*t;
  y = v0*sin(theta)*t-0.5*g*t*t;
  X = ct.getX(x);
  Y = ct.getY(y);
  ellipse(X,Y,w,w);
  t = t + 0.05;
  if( ct.outofRange() ) t = 0;
  println("x = "+x+" m  y = "+y+" m");
}
```

Example 9

```
// Drawing random data
float x,y;
int X1,Y1,X2,Y2;
Transform ct;

void setup() {
  size(700,200);
  ct = new Transform(0,1000, 0,20);
  stroke(0,255,0);
  X1=X2=Y1=Y2=0;
  x=y=0;
  background(0);
}
void draw(){
  y = 10 + random(-5,5);
  X1 = X2;
  Y1 = Y2;
  y = 10 + random(-5,5);
  x += 4;
  X2 = ct.getX(x);
  Y2 = ct.getY(y);
  line(X1,Y1, X2,Y2);

  if( ct.outofRange() ) {
    x = 0;
    background(0);
    X2 = ct.getX(0);
    Y2 = ct.getY(10);
  }
  println("x = "+x+" m  y = "+y+" m");
}
```



Example 10

```
// Reading serial data

import processing.serial.*;

Serial myPort;  // The serial port

void setup() {
  // List all the available serial ports
  println(Serial.list());
  // Open the port you are using  at the rate you want:
  myPort = new Serial(this, Serial.list()[0], 9600);
}

void draw() {
  while (myPort.available() > 0) {
    int inByte = myPort.read();
    println(inByte);
  }
}
```

Example 11

```
// Writing data to a file

PrintWriter output;

void setup() {
  size(200,200);
  background(255);
  output = createWriter("data.txt");
}

void draw() {
}

void mousePressed() {
  fill(0);
  rectMode(CENTER);
  rect(mouseX,mouseY,16,16); // Draw rectangle
  output.println(mouseX + "," + mouseY);  // Write to the file
  output.flush(); // Writes the remaining data to the file
}

void stop() {
  output.close();  // Finishes the file
  exit();
}
```

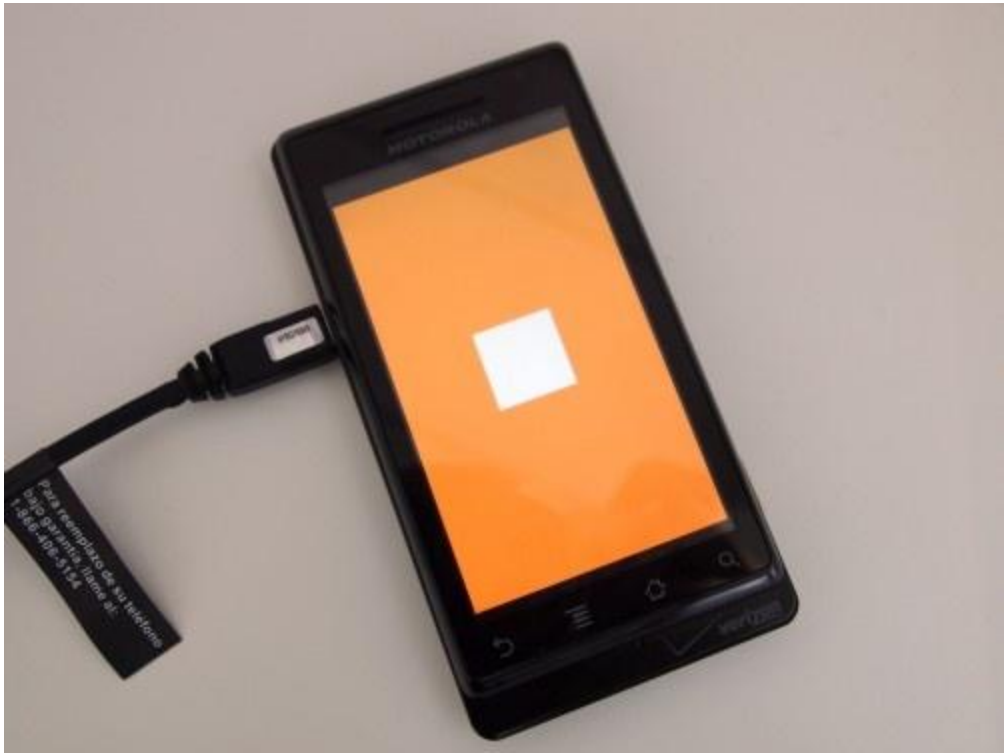
Example 12: Date Time Functions

- millis()
- second()
- minute()
- hour()
- day()
- month()
- year()

```
// Writing date/time data

void setup() {
  size(300,300);
  noLoop();
}

void draw() {
  int d = day();     // Values from 1 - 31
  int m = month();   // Values from 1 - 12
  int y = year();    // 2011, 2012, 2013, etc.
  int S = second(); // 0-59
  String s = String.valueOf(d); text(s, 10, 20);
  s = String.valueOf(m);        text(s, 10, 40);
  s = String.valueOf(y);        text(s, 10, 60);
  s = String.valueOf(S);        text(s, 10, 80);
}
```

Android Mode

- See: http://processing.org/tutorials/android/



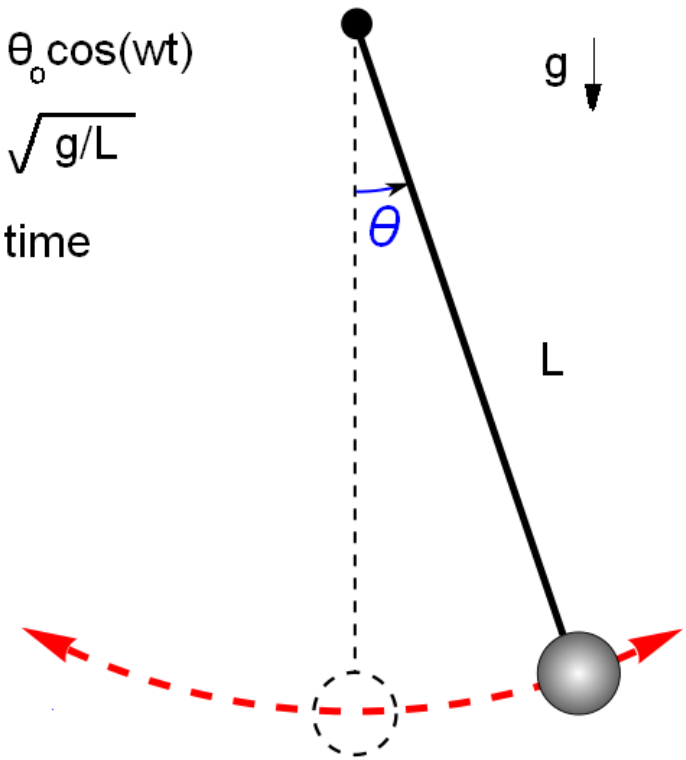
Homeworks

1. Write a program to simulate and animate the motion of a simple pendulum. The initial angle (θ_0) of the pendulum should be given by using the mouse.

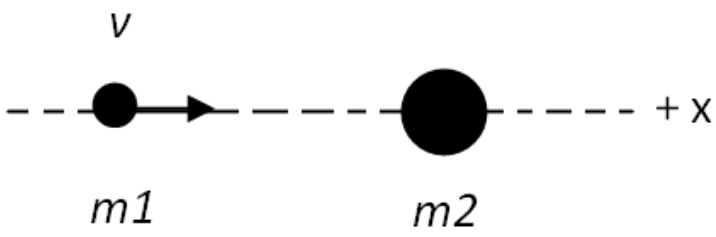
$\theta = \theta_o \cos(wt)$

$w = \sqrt{g/L}$

t = time



2. A ball of mass m1 and velocity v moves in the direction of +x and makes a central elastic collision with another ball, which is initially at rest and has mass m2, as shown in Figure. Write a program, to perform a simulation and animation of the system for the given m1, m2, and v.



3. Repeat Problem 1 or 2 in Android mode.
(Load your program to a phone having Android OS and run it).