

マイコンをはじめよう

第6回 ProcessingからLEDを調光する

川上 博

2013/08/31

今日のテーマ

ProcessingからRGBledを調光する

cosine 関数を描く : Example 601P

赤色の明るさを変化させる : Example 602P

グラフに目盛りを付ける : Example 603P

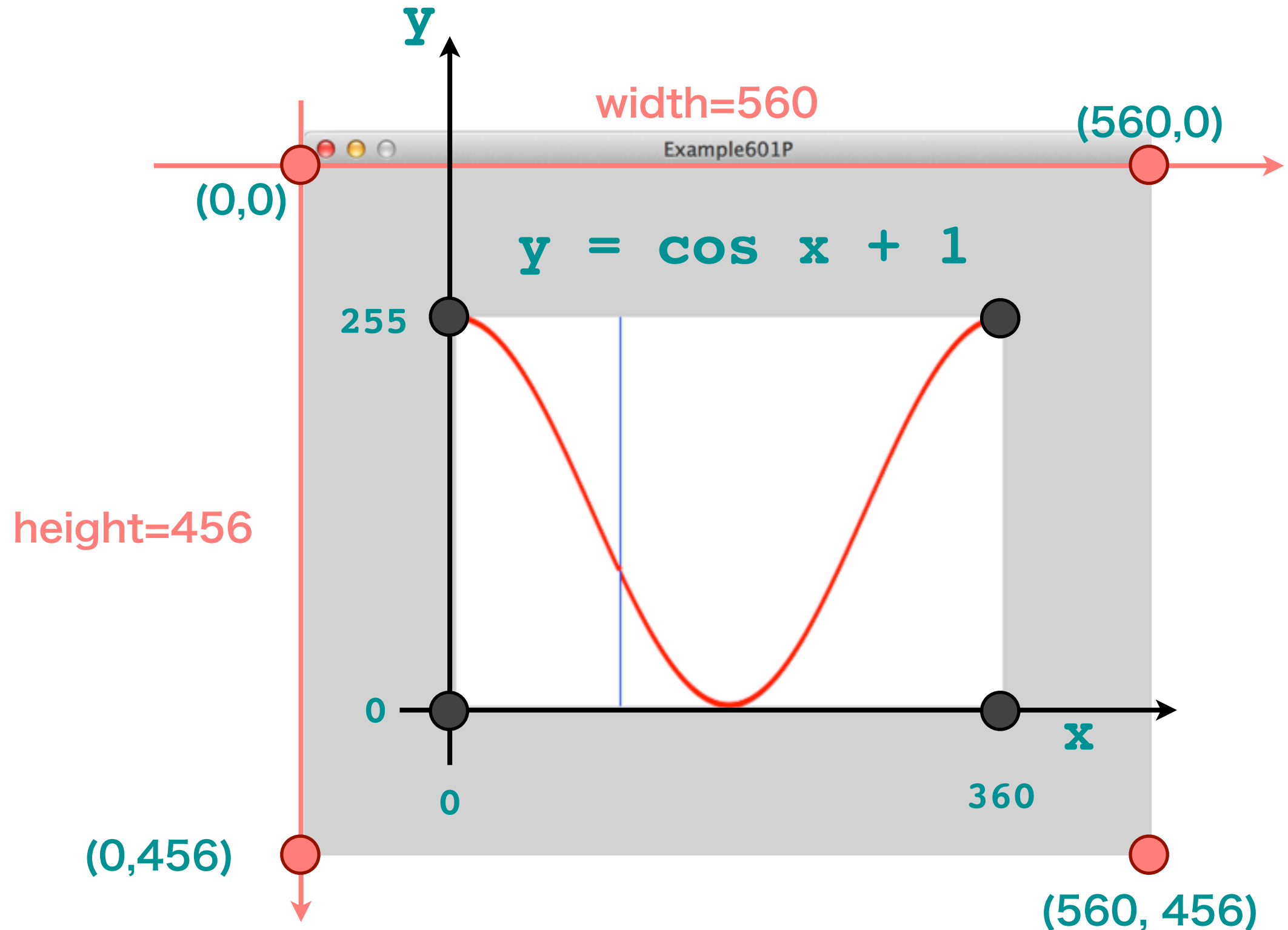
RGBから色相(hue)をつくる : Example 604P

RGB値をArduinoに送る : Example 605P

RGBledを調光する : Example 605A

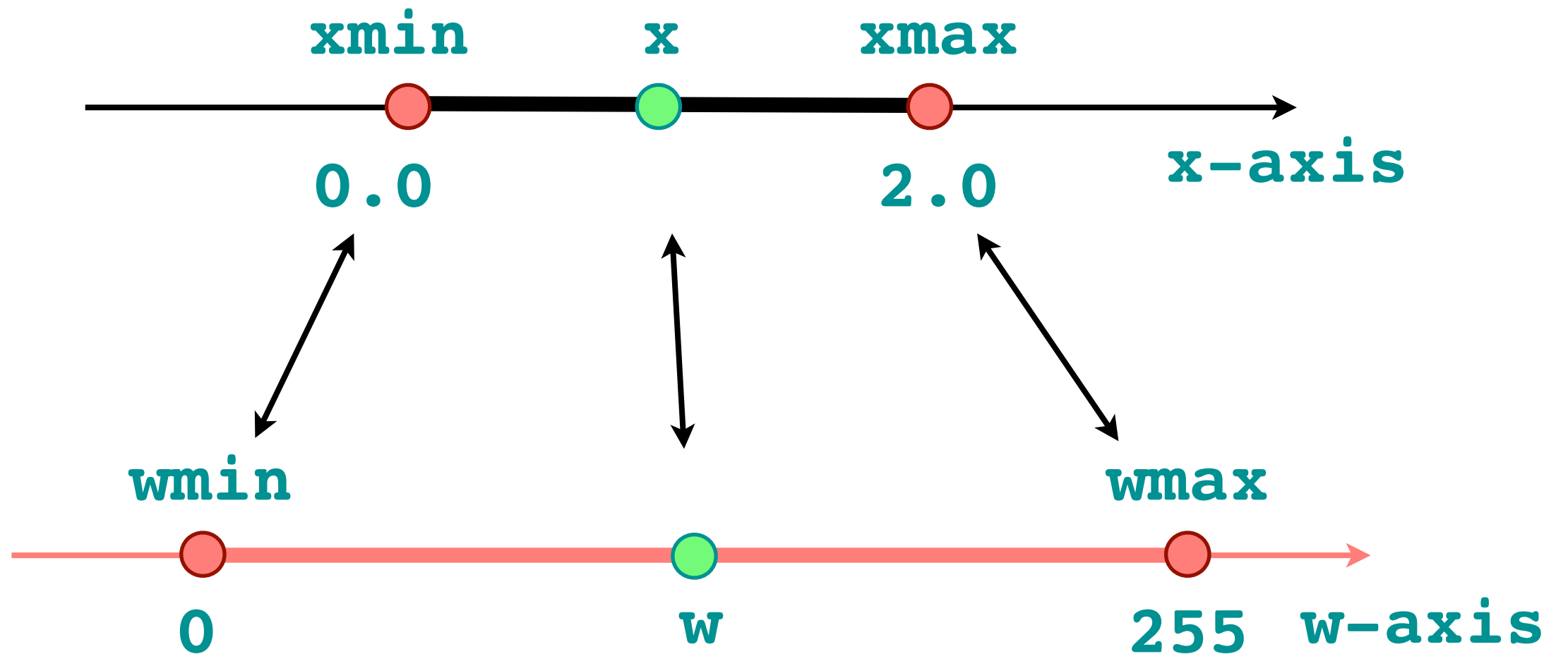
cosine 関数を描く : Example 601P

関数のグラフを描く



データのスケーリング

$$w = \text{map}(x, \text{xmin}, \text{xmax}, \text{wmin}, \text{wmax})$$

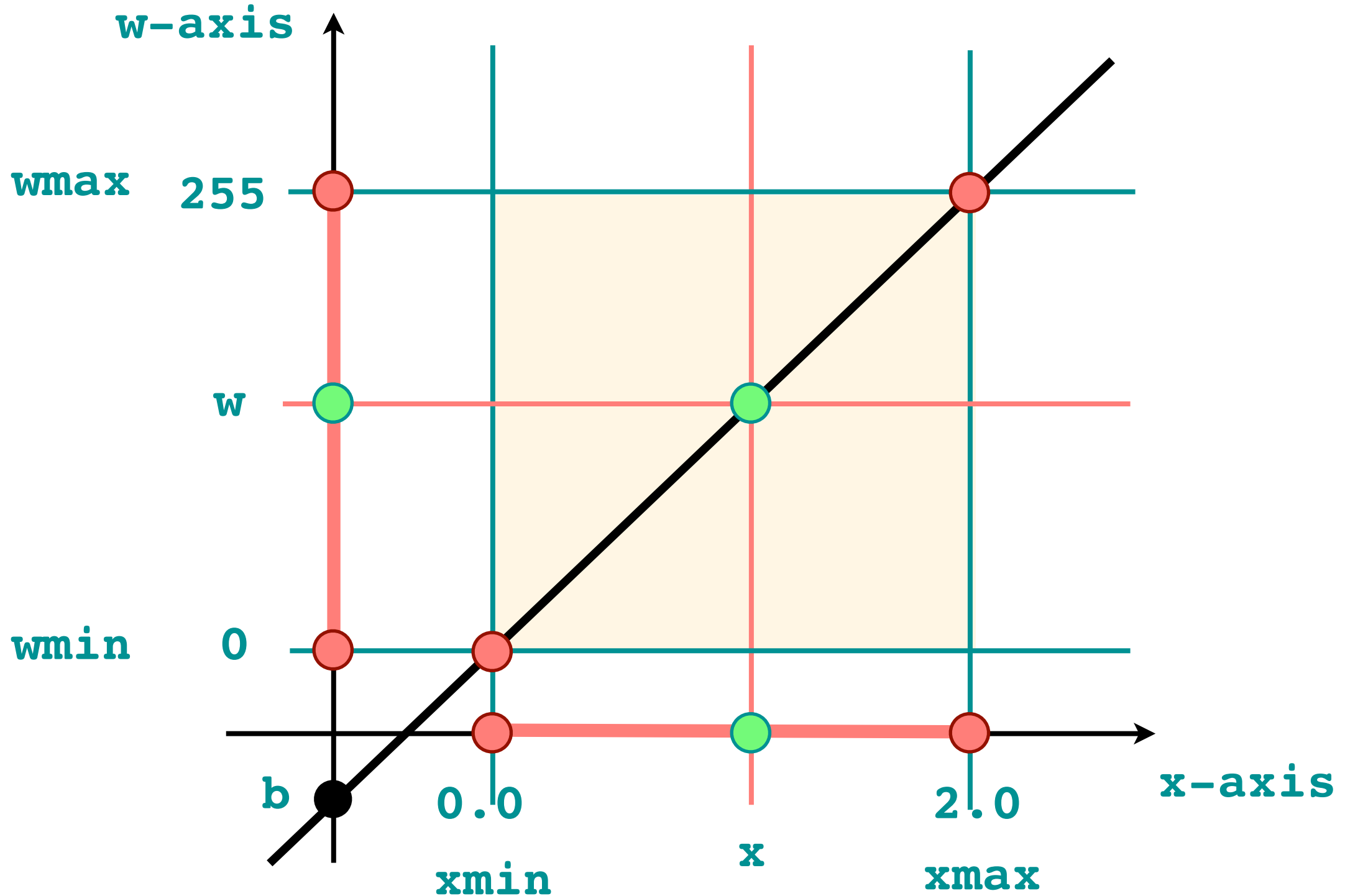


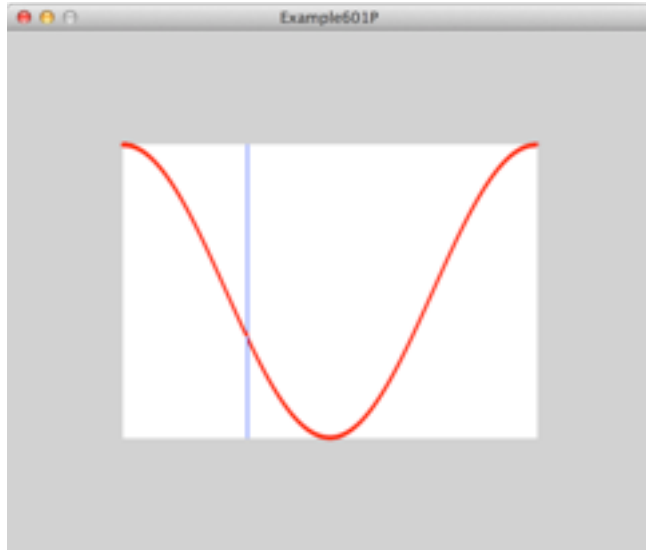
6

$$w = ax + b$$

ここに $a = (w_{\max} - w_{\min}) / (x_{\max} - x_{\min}) ;$

$$b = (w_{\min} * x_{\max} - w_{\max} * x_{\min}) / (x_{\max} - x_{\min})$$





```
// Example 601P
int N=360,M=256;
int xmarginL=100, xmarginR=100;
int ymarginU=100, ymarginL=100;
float h=TWO_PI/N;
float x, val, wx;
float y,wy;
```

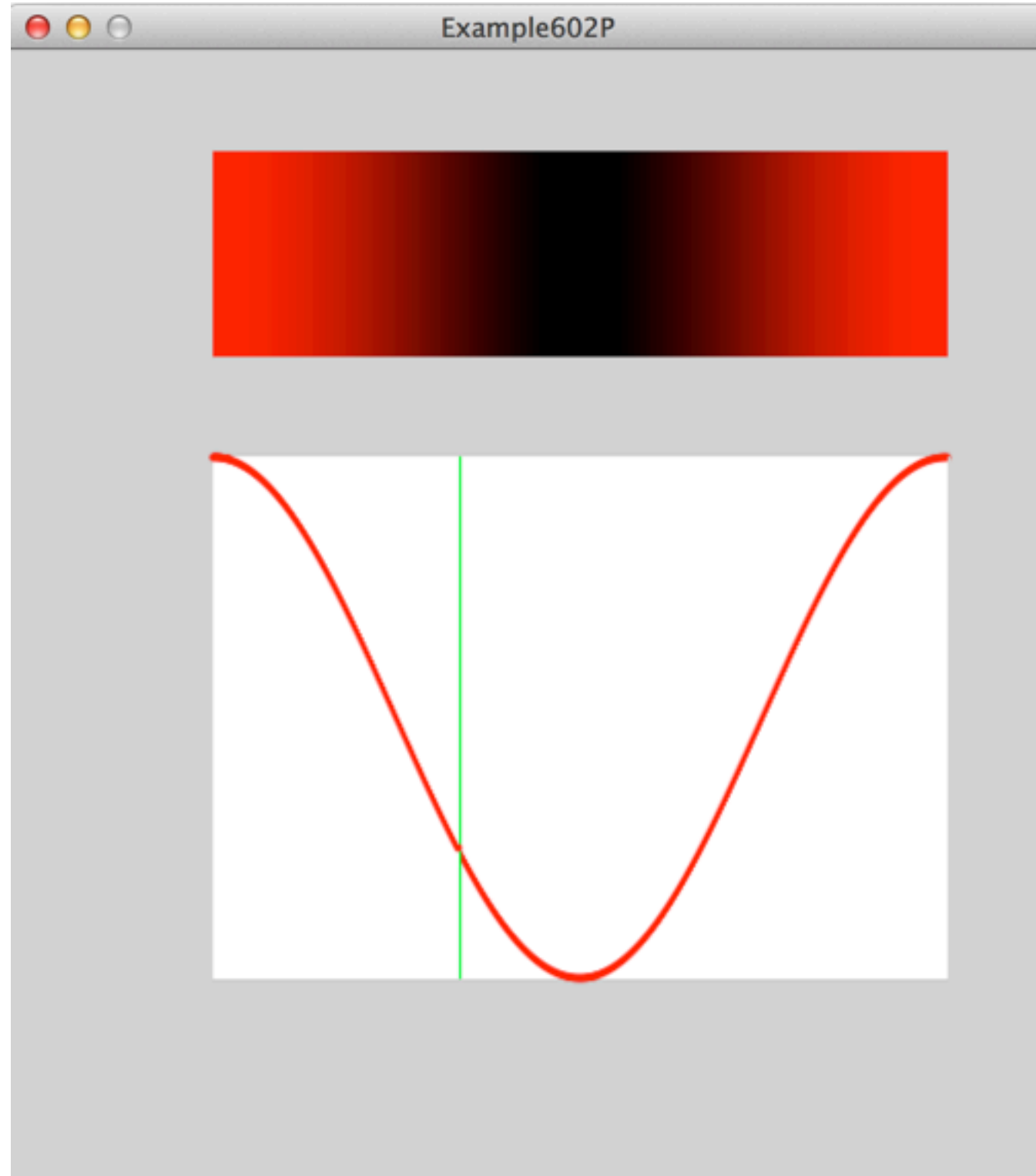
```
void setup(){
  size(N+xmarginL+xmarginR, M+ymarginU+ymarginL);
  background(200);
  frameRate(30);
  noStroke();
  rectMode(CORNERS);
  rect(xmarginL, ymarginU, width-xmarginR, height-ymarginL);
  x=0.0;
}
```

```
void draw(){
  val=h*x;
  y = cos(val);
  wx = map(val, 0.0, TWO_PI, xmarginL, width-xmarginR);
  wy = map(y, -1.0, 1.0, height-ymarginL, ymarginU);

  strokeWeight(1);
  stroke(255);
  line(wx, ymarginU, wx, height-ymarginL); // white line
  stroke(0, 0, 255);
  line(wx+1, ymarginU, wx+1, height-ymarginL); // blue line

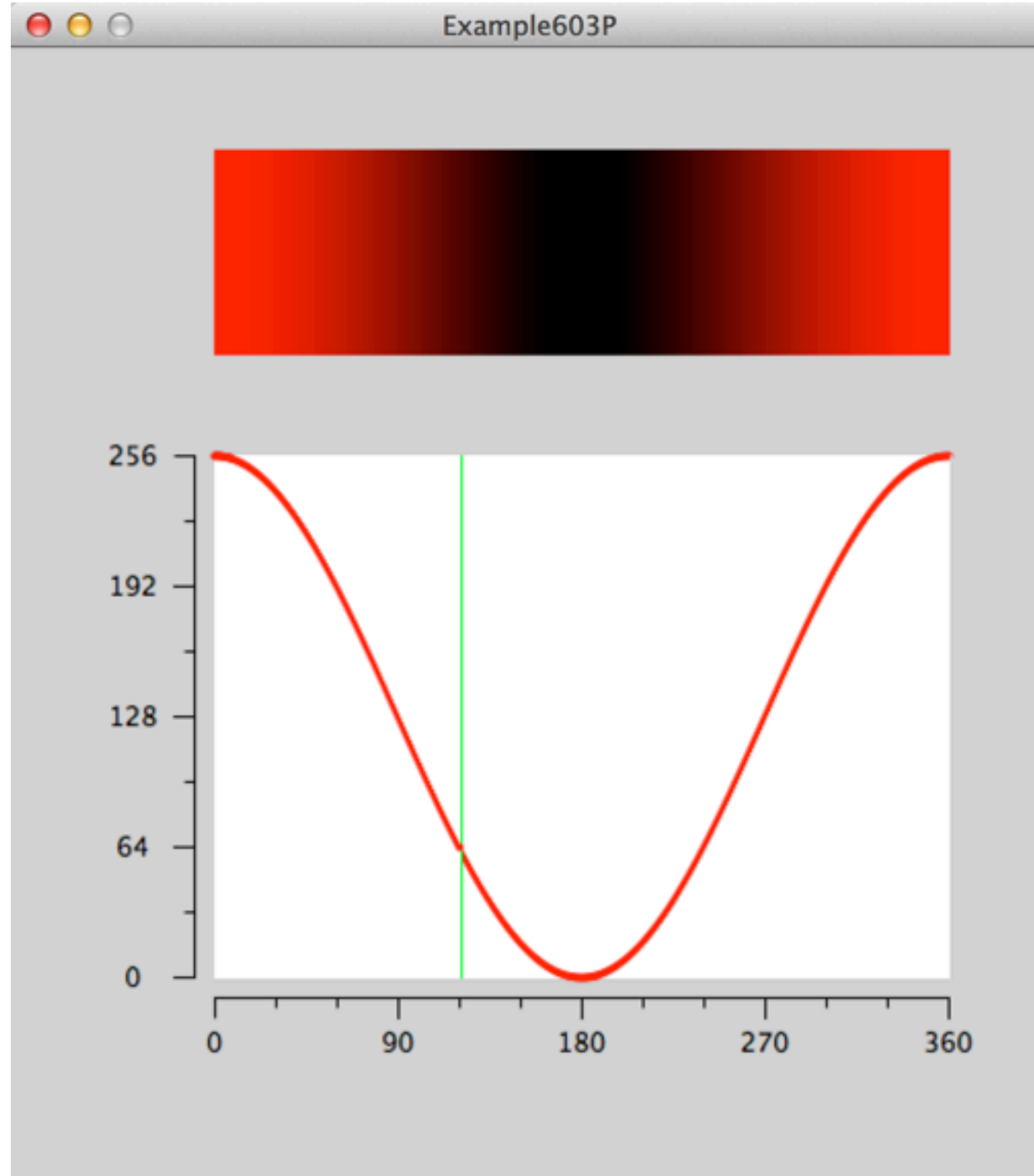
  strokeWeight(4);
  stroke(255,0,0);
  point(wx,wy);
  x++;
  if (x > N) {
    strokeWeight(1);
    stroke(200);
    line(wx+1, ymarginU, wx+1, height-ymarginL); // final line
    x = 0.0;
  }
}
```

赤色の明るさを変化させる : Example 602P
グラフに目盛りを付ける : Example 603P
RGBから色相(hue)をつくる : Example 604P



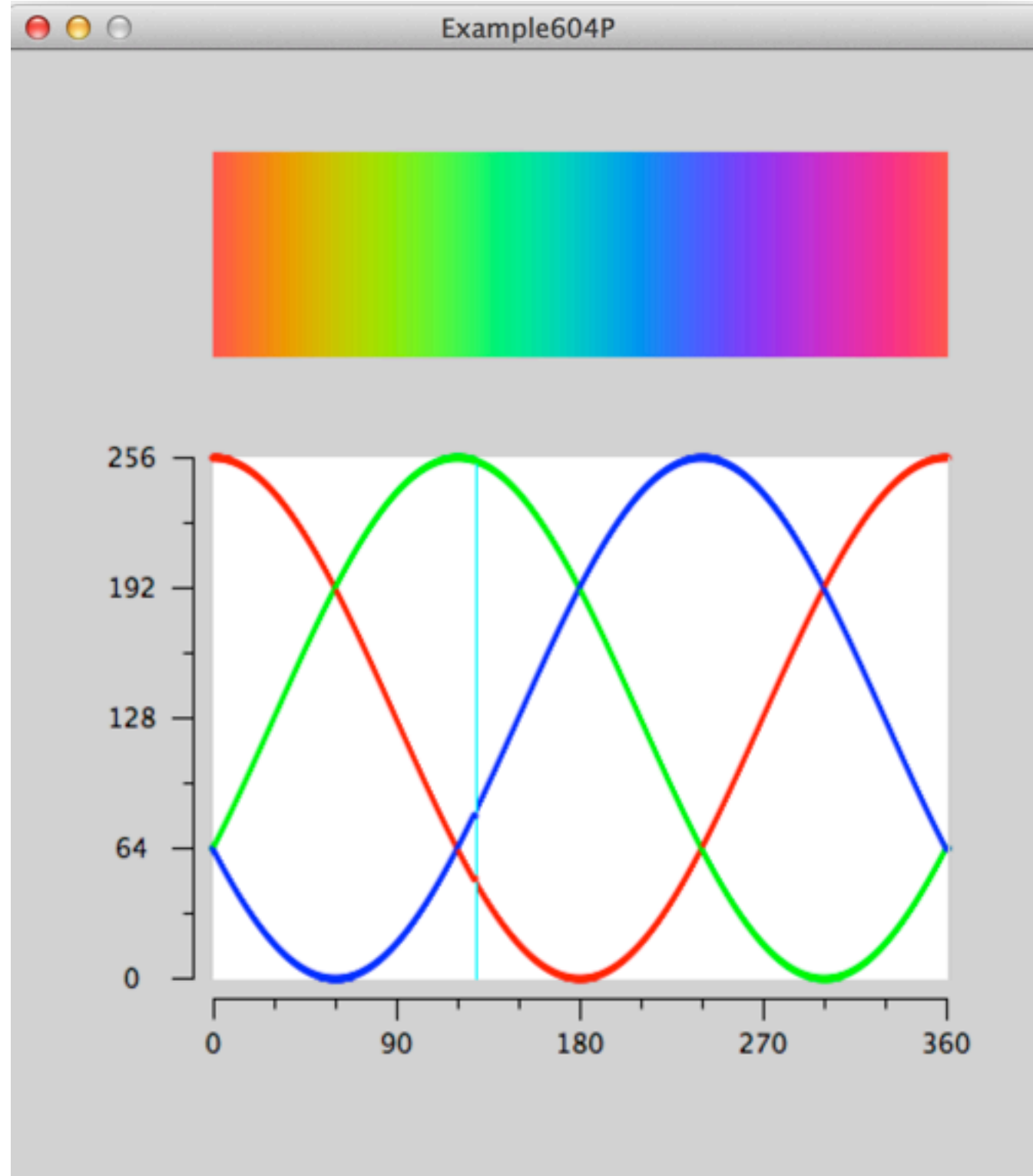
```
void draw(){
    val=h*x;
    y1 = cos(val);
    wx = map(val, 0.0, TWO_PI,xmarginL, width-xmarginR);
    wy1 = map(y1, -1.0, 1.0, height-ymarginL,ymarginU);
    r = map(y1, -1.0, 1.0, 0, 255);
    strokeWeight(1);
    stroke(r,0,0);
    line(wx, 50, wx, 150);
    stroke(255);
    line(wx, ymarginU, wx, height-ymarginL); // white line
    stroke(0,255,0);
    line(wx+1, ymarginU, wx+1, height-ymarginL); // green line

    strokeWeight(4);
    stroke(255,0,0);
    point(wx,wy1);
    x++;
    if (x > N) {
        strokeWeight(1);
        stroke(200);
        line(wx+1, ymarginU, wx+1, height-ymarginL); // white line
        x = 0.0;
    }
}
```



```
void axes(){
  stroke(0);fill(0);textAlign(CENTER, BOTTOM);
  line(xmarginL-10,ymarginU,xmarginL-10,height-ymarginL); // y-axis
  for(int i=ymarginU; i<=height-ymarginL; i+=32){
    line(xmarginL-15,i,xmarginL-10,i);
  }
  for(int i=ymarginU; i<=height-ymarginL; i+=64){
    line(xmarginL-20,i,xmarginL-10,i);
    text(256-(i-ymarginU), xmarginL-40, i+8);
  }

  line(xmarginL,height-ymarginL+10, width-xmarginR, height-ymarginL+10); // x-axis
  for(int i=xmarginL; i<=width-xmarginR; i+=30){
    line(i,height-ymarginL+10,i,height-ymarginL+15);
  }
  for(int i=xmarginL; i<=width-xmarginR; i+=90){
    line(i,height-ymarginL+10,i,height-ymarginL+20);
    text(i-xmarginL, i, height-ymarginL+40);
  }
}
```



```
void draw(){
    val=h*x;
    y1 = cos(val);
    y2 = cos(val-TWO_PI/3.0);
    y3 = cos(val+TWO_PI/3.0);
    wx = map(val, 0.0, TWO_PI,xmarginL, width-xmarginR);
    wy1 = map(y1, -1.0, 1.0, height-ymarginL,ymarginU);
    wy2 = map(y2, -1.0, 1.0, height-ymarginL,ymarginU);
    wy3 = map(y3, -1.0, 1.0, height-ymarginL,ymarginU);

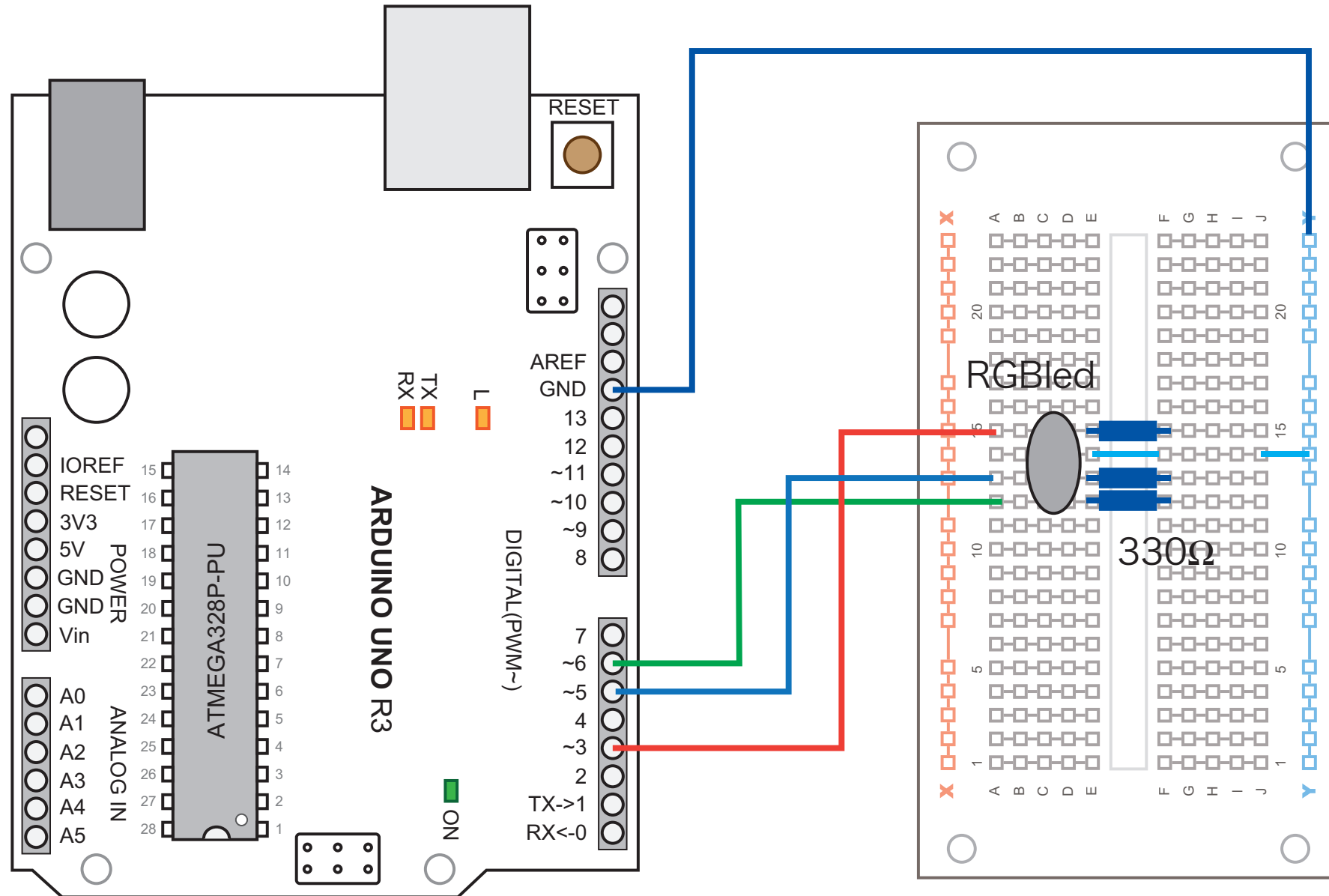
    r = map(y1, -1.0, 1.0, 0, 255);
    g = map(y2, -1.0, 1.0, 0, 255);
    b = map(y3, -1.0, 1.0, 0, 255);

    strokeWeight(1);
    stroke(r,g,b);
    line(wx, 50, wx, 150);
    stroke(255);
    line(wx, ymarginU, wx, height-ymarginL); // white line
    stroke(0,255,255);
    line(wx+1, ymarginU, wx+1, height-ymarginL); // green line

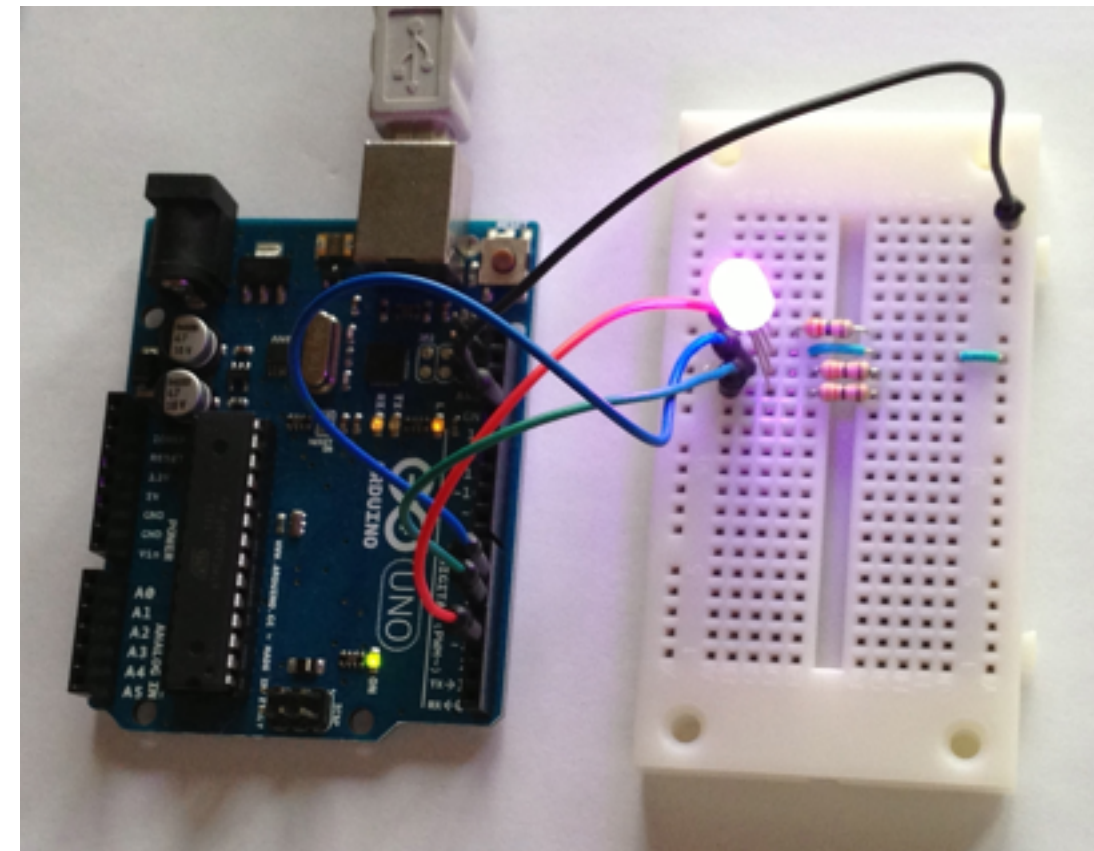
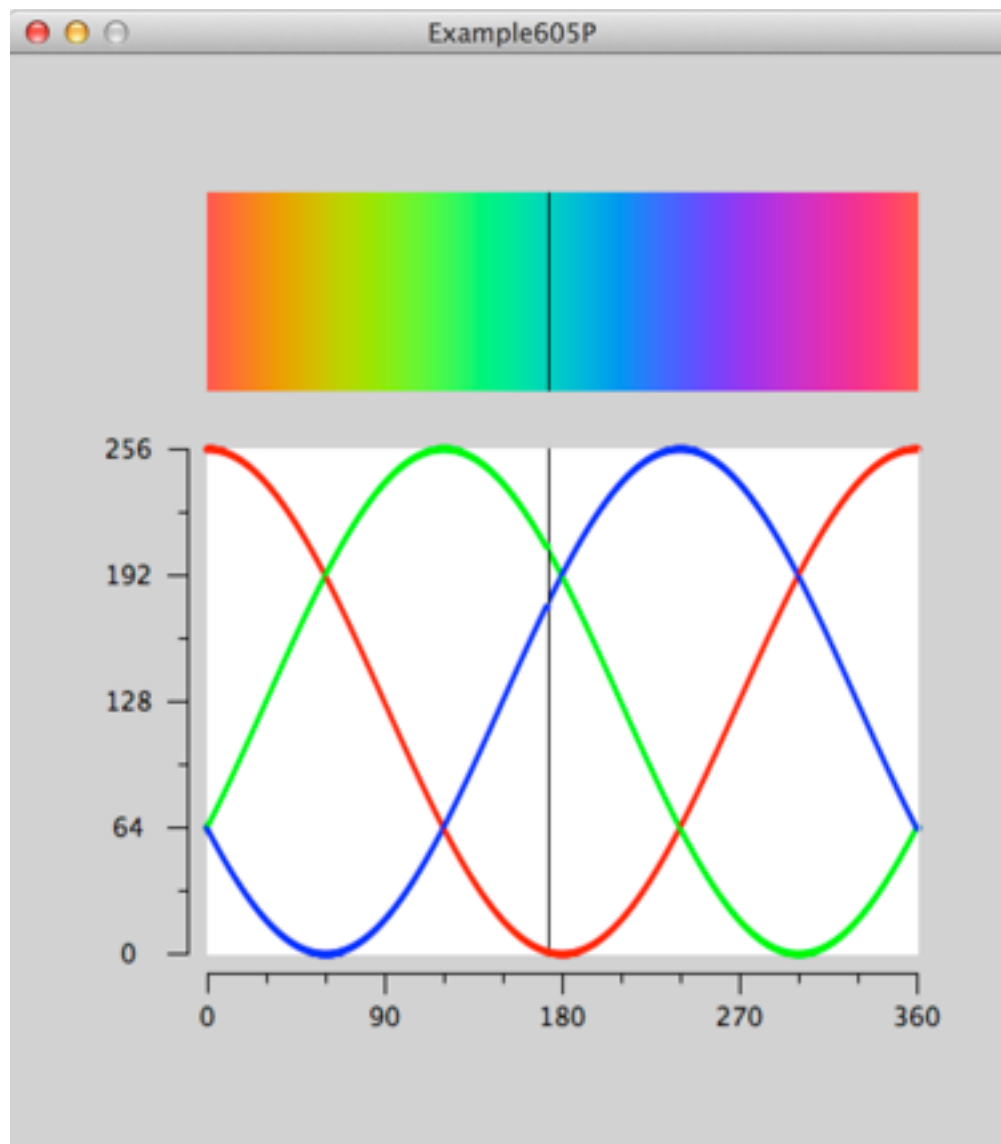
    strokeWeight(4);
    stroke(255,0,0);
    point(wx,wy1);
    stroke(0,255,0);
    point(wx,wy2);
    stroke(0,0,255);
    point(wx,wy3);
    x++;
    if (x > N) {
        strokeWeight(1);
        stroke(200);
        line(wx+1, ymarginU, wx+1, height-ymarginL); // white line
        x = 0.0;
    }
}
```

Processingから調光する

RGBledを点灯する



RGBledを点灯する



```
// Example 605A
// Serial communications:
// Receiving 4 bytes data from Processing
// Coded by H. Kawakami July 16, 2013

const char HEADER='H';    //****
const int TOTAL_BYTES=4; //****
const int ledB = 6;
const int ledG = 5;
const int ledR = 3;
byte b, g, r;            //****

void setup(){
  Serial.begin(9600);    //****
  pinMode(ledB, OUTPUT);
  pinMode(ledG, OUTPUT);
  pinMode(ledR, OUTPUT);
}

void loop() {
  if (Serial.available() >= TOTAL_BYTES) { //****
    if(Serial.read()==HEADER){ //****
      r = Serial.read(); //****
      g = Serial.read(); //****
      b = Serial.read(); //****
    } //****
  } //****
  analogWrite(ledB, b);
  analogWrite(ledG, g);
  analogWrite(ledR, r);
}
```

```
// Example 605P
// Serial communications:
// Sending 4 bytes data to Arduino
// Coded by H. Kawakami July 16, 2013
//
```

```
import processing.serial.*; //****
Serial port = new Serial(this, Serial.list()[5], 9600); //****

public static final char HEADER='H'; //****
int N=360, M=256, hmin=70, hmax=170;
int xmarginL=100, xmarginR=50;
int ymarginU=200, ymarginL=100;
float h=TWO_PI/N;
float x, val, wx;
float y1,wy1,y2,wy2,y3,wy3;
float r, g, b;
```

```
void draw(){
    val=h*x;
    y1 = cos(val);
    y2 = cos(val-TWO_PI/3.0);
    y3 = cos(val+TWO_PI/3.0);
    wx = map(val, 0.0, TWO_PI,xmarginL, width-xmarginR);
    wy1 = map(y1, -1.0, 1.0, height-ymarginL,ymarginU);
    wy2 = map(y2, -1.0, 1.0, height-ymarginL,ymarginU);
    wy3 = map(y3, -1.0, 1.0, height-ymarginL,ymarginU);
    r = map(y1, -1.0, 1.0, 0, 255);
    g = map(y2, -1.0, 1.0, 0, 255);
    b = map(y3, -1.0, 1.0, 0, 255);
    port.write(HEADER); //****
    port.write(int(r)); //****
    port.write(int(g)); //****
    port.write(int(b)); //****
    strokeWeight(1);
    stroke(r,g,b);
    line(wx, hmin, wx, hmax);
    stroke(0);
    line(wx+1, hmin, wx+1, hmax);
    stroke(255);
    line(wx, ymarginU, wx, height-ymarginL); // white line
    stroke(0);
    line(wx+1, ymarginU, wx+1, height-ymarginL); // green line

    strokeWeight(4);
    stroke(255,0,0);
    point(wx,wy1);
    stroke(0,255,0);
    point(wx,wy2);
    stroke(0,0,255);
    point(wx,wy3);
    x++;
    if (x > N) {
        strokeWeight(1);
        stroke(200);
        line(wx+1, hmin, wx+1, hmax);
        line(wx+1, ymarginU, wx+1, height-ymarginL); // white line
        x = 0.0;
    }
}
```

```
void axes(){
  stroke(0);fill(0);textAlign(CENTER, BOTTOM);
  line(xmarginL-10,ymarginU,xmarginL-10,height-ymarginL); // scale for y-axis
  for(int i=ymarginU; i<=height-ymarginL; i+=32){
    line(xmarginL-15,i,xmarginL-10,i);
  }
  for(int i=ymarginU; i<=height-ymarginL; i+=64){
    line(xmarginL-20,i,xmarginL-10,i);
    text(256-(i-ymarginU), xmarginL-40, i+8);
  }

  line(xmarginL,height-ymarginL+10, width-xmarginR, height-ymarginL+10); // scale for x-axis
  for(int i=xmarginL; i<=width-xmarginR; i+=30){
    line(i,height-ymarginL+10,i,height-ymarginL+15);
  }
  for(int i=xmarginL; i<=width-xmarginR; i+=90){
    line(i,height-ymarginL+10,i,height-ymarginL+20);
    text(i-xmarginL, i, height-ymarginL+40);
  }
}

void setup(){
  size(N+xmarginL+xmarginR, M+ymarginU+ymarginL);
  background(200);
  frameRate(10);
  noStroke();
  rectMode(CORNERS);
  rect(xmarginL,hmin,width-xmarginR,hmax);
  rect(xmarginL, ymarginU, width-xmarginR, height-ymarginL);
  axes();
  x=0.0;
}
```