Blowing the Light Bulb on/off – ARDUINO CODE

```
/*
 Blowing the Light Bulb on/off

 Uses two fans as input devices and calculates the speed of their turning
 in rate per minute (rpm). The difference of their rpm is translated into
 a value in between 0 to 255 that serves to adjust the brightness of an LED.

 The set-up:
 * two standard PC-Fans (80mm)
 * two LEDs and two 10K Ohm resistors (Arduino input fan circuits)
 * an LED and a 100 Ohm resistor (Arduino output circuit)
 * a manual dimmer for light bulbs and a plug socket adapter
 * a photo cell
 * a light bulb

 The Arduino circuits:
 * Fans' signal wires (yellow) connected to digital pins 2 / 3 (external interrupts)
 * Fans' ground wires (black) connected to ground
 * Fans' power connections (red wire) not connected
 * LED and 10K Ohm resistor connected in line to signal pin and 5V
 * LED and 100 Ohm resistor connected in line to analog out pin 11 and ground

 The dimmer circuit:
 * photo cell attached to potentiometer of the dimmer, short-circuiting it

 Attention:
 Photo cell needs to be arranged next to output LED but MUST NOT BE CONNECTED with it in any
way!
 The arrangement of photo cell and output LED needs to be shaded from ambient light.

 Created 30 September 2009
 By Katja Knecht & Yuan Qi

 Reference:
 Reading Fan RPM
 http://www.arduino.cc/playground/Main/ReadingRPM

 */


// VARIABLES

volatile byte rpmcount1; // counter turns of fan 1
volatile byte rpmcount2; // counter turns of fan 2
unsigned int rpm1; // stores the rpm for fan 1
unsigned int rpm2; // stores the rpm for fan 2
unsigned long timeold; // stores the time value of the previous turn
int rpmdiff; // stores the difference in between rpm1 and rpm2
int brightness; // stores the brightness of the LED
int ledPin =  11; // LED connected to PWM pin 11
int rpmthreshold = 1000; // defines rpm threshold for max and min brightness
int initbrightness = 50; // defines initial brightness of the light
int minbrightness = 0; // defines min brightness of the light
int maxbrightness = 100; // defines max brightness of the light


// SETUP

void setup()
{
  Serial.begin(9600); // opens serial port, sets data rate to 9600 bps
  attachInterrupt(0, rpm_fun1, RISING);
   /* calls function rpmfun in case of an external interupt (= fan signal) on digital pin 2
   (number 0); triggers when the pin goes from low to high*/
  attachInterrupt(1, rpm_fun2, RISING);
   /* calls function rpmfun in case of an external interupt (= fan signal) on digital pin 3
   (number 1); triggers when the pin goes from low to high*/
  pinMode(ledPin, OUTPUT); // initialize the PWM pin as an output
  rpmcount1 = 0; // sets value of variable to 0
  rpmcount2 = 0; // sets value of variable to 0
  rpmdiff = 0; // sets value of variable to 0
  rpm1 = 0; // sets value of variable to 0
  rpm2 = 0; // sets value of variable to 0
  brightness = initbrightness; // sets value of variable to 127
}


// LOOP

void loop()
{
  if (rpmcount1 >= 20 || rpmcount2 >=20) { // execute statements if either of the fans has
                                           //turned ten times
    //Update RPM every 20 counts, increase this for better RPM resolution,
    //decrease for faster update

    rpm1 = 30*1000/(millis() - timeold)*rpmcount1; // calculates the rpm of fan 1
    rpm2 = 30*1000/(millis() - timeold)*rpmcount2; // calculates the rpm of fan 2
    timeold = millis(); // stores the current time as reference for next turn
    rpmdiff = rpm1-rpm2; // calculates the difference of rpms; positive if fan 1 rotates
                         //faster than fan 2; negative if fan 2 rotates faster

    if (rpmdiff >0){ // if fan 1 rotates faster than fan 2 LED lights up
      if (rpmdiff <= rpmthreshold){ // if value is in between 0 and threshold, brightness is
                                    // mapped to value
        brightness = map(rpmdiff, 0, rpmthreshold, initbrightness, maxbrightness); // maps
                                                                                   // values
      }
      else if (rpmdiff > rpmthreshold) { // if value is higher than threshold brightness is
                                         // set to max
        brightness = maxbrightness; // LED on
      }
    }

    else if (rpmdiff < 0){ // if fan 2 rotates faster than fan 1 LED dims down
      if (rpmdiff >= -rpmthreshold){ // if value is in between negative threshold and 0,
                                     // brightness is mapped to value
        brightness = map(rpmdiff, -rpmthreshold, 0, minbrightness, initbrightness); // maps
                                                                                    // values
      }
      else if (rpmdiff < -rpmthreshold){ // if value is smaller than threshold, brightness is
                                         // set to zero
        brightness =minbrightness; // LED off
      }
    }

    else { // statement executed if none of the above mentioned conditions apply
      brightness = initbrightness; // sets brightness to initial value
    }

    analogWrite(ledPin, brightness); // writes the value of brightness (in between 0 and 255)
                                     // to PWM pin
    rpmcount1 = 0; // resets counter
    rpmcount2 = 0; // resets counter

    // PRINT VALUES IN SERIAL MONITOR FOR DEBUGGING PURPOSES
    Serial.print(rpm1,DEC); // prints the decimal value of rpm1 in an ASCII string
    Serial.print("  "); // prints spaces
    Serial.print(rpm2, DEC); // prints the decimal value of rpm2 in an ASCII string
    Serial.print("  "); // prints spaces
    Serial.println(brightness, DEC); // prints the decimal value of brightness in an ASCII
                                     // string
  }

  else if ((millis() - timeold) > 2000){ // if the brightness of the LED has not been changed
                                         // for 2 seconds reset brightness
    brightness = initbrightness; // sets brightness to initial value
  }
}


// FUNCTIONS

void rpm_fun1()
{
  rpmcount1++;
  //Each rotation, this interrupt function is run twice
}
void rpm_fun2()
{
  rpmcount2++;
  //Each rotation, this interrupt function is run twice
}
```