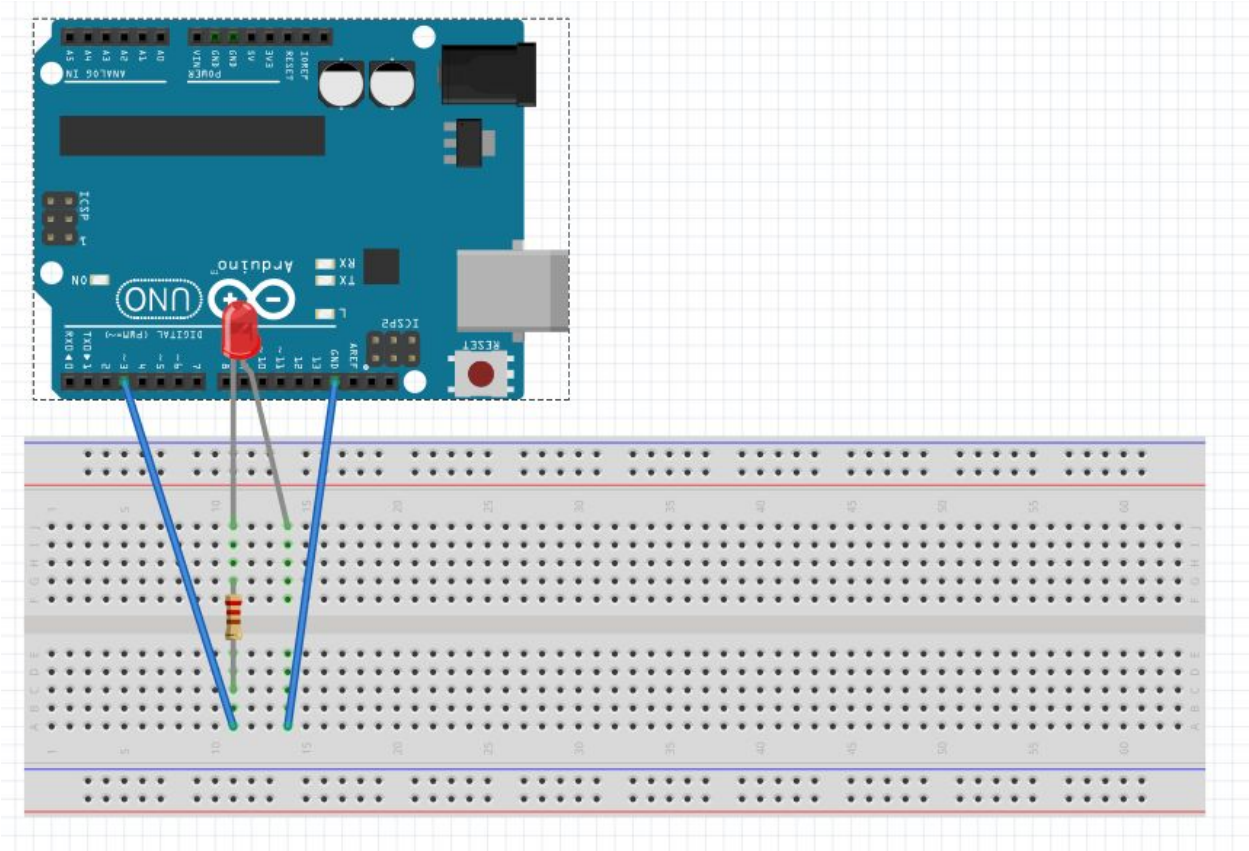
Group No: 15

Manvendra Singh, Rohan Verma, Sarthak Mittal

Question 1

Circuit Diagram



Arduino Code

```
 // Read data from the serial and turn ON or OFF a light depending on
the value
```

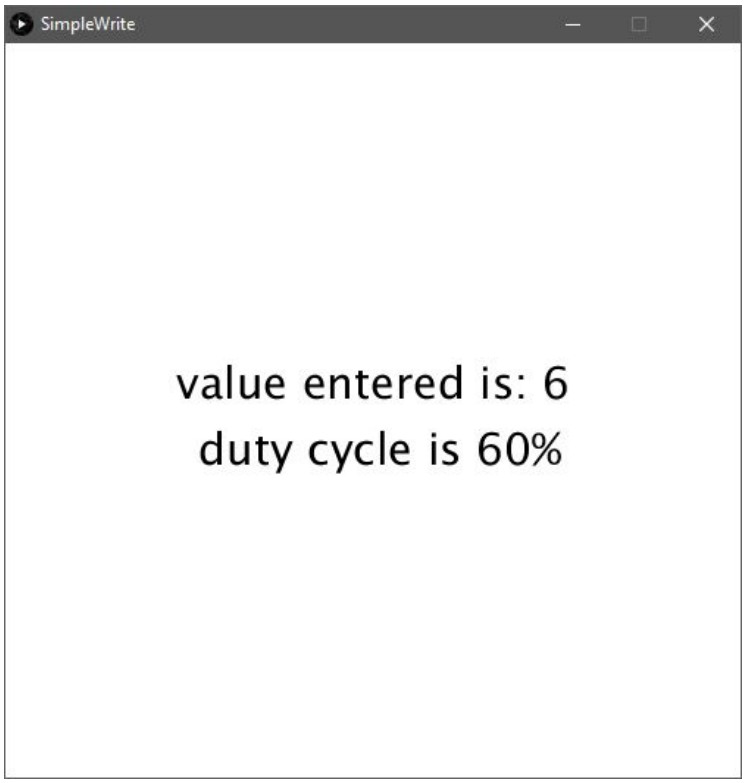
```
char val; // Data received from the serial port
int ledPin = 3; // Set the pin to 3
int pwm_intensity = 255;
void setup() {
  pinMode(ledPin, OUTPUT); // Set pin as OUTPUT
  Serial.begin(9600); // Start serial communication at 9600 bps
}

void loop() {
  while (Serial.available()) { // If data is available to read,
    val = Serial.read(); // read it and store it in val
  }

  int intensity = val - '0';

  if(intensity >=0 && intensity < 10){
    pwm_intensity = (int)((intensity / 9.0)*255.0);
  }

  analogWrite(ledPin, pwm_intensity);


  delay(100); // Wait 100 milliseconds for next reading
}
```

Processing Code

```
import processing.serial.*;

Serial myPort;  // Create object from Serial class
int val;        // Data received from the serial port

String myText = "Enter value between 0-9";
String input_value = "";
```

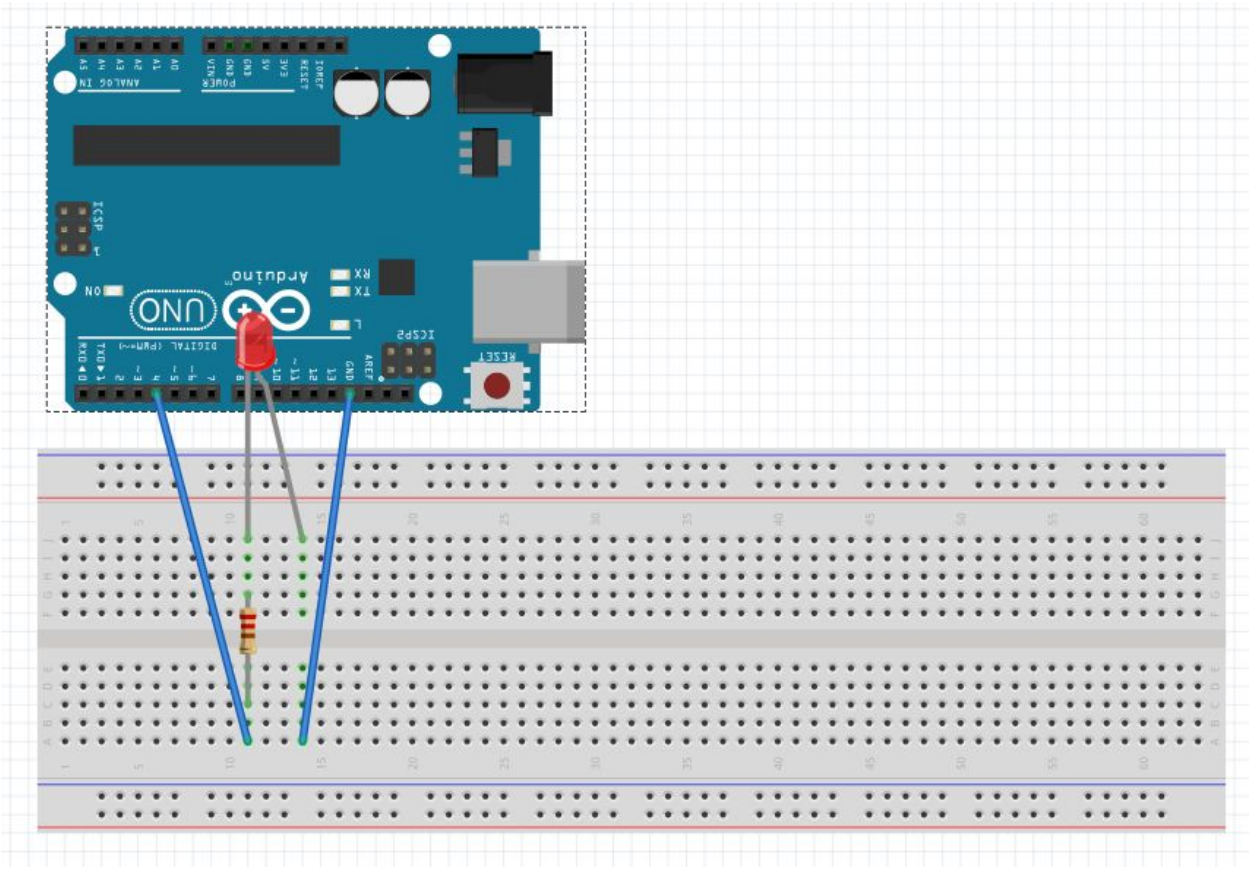
```
void setup()
{
  size(500, 500);
  textAlign(CENTER, CENTER);
  textSize(30);
  fill(0);

  // setting up serial com
  String portName = "COM3";
  myPort = new Serial(this, portName, 9600);
}

void draw() {
  background(255);

  text(myText, 0, 0, width, height);
  //rect(50, 50, 100, 100);      // Draw a square
}

void keyPressed() {
  if (keyCode == ENTER) {

      int duty_cycle = (int)((Integer.parseInt(input_value)*10.0));
      myText = "value entered is: " + input_value + "\n duty cycle is
"+ duty_cycle + "%";
      myPort.write(input_value);
      input_value = "";
  }

  // append any user input to string
  else if (keyCode != SHIFT && keyCode != CONTROL && keyCode != ALT) {
      input_value = input_value + key;
  }
}
```

Screenshots:



Question 2:

Circuit Diagram



Arduino Code

```
 // Read data from the serial and turn ON or OFF a light depending on the value

 char val; // Data received from the serial port
 int ledPin = 4; // Set the pin to 3

 void setup() {
   pinMode(ledPin, OUTPUT); // Set pin as OUTPUT
```

```
  Serial.begin(9600); // Start serial communication at 9600 bps
}

void loop() {
  while (Serial.available()) { // If data is available to read,
    val = Serial.read(); // read it and store it in val
  }


  if(val == 'H'){
    digitalWrite(ledPin, HIGH);
  }
  else if(val == 'L'){
    digitalWrite(ledPin, LOW);
  }


  delay(100); // Wait 100 milliseconds for next reading
}
```

Processing Code:

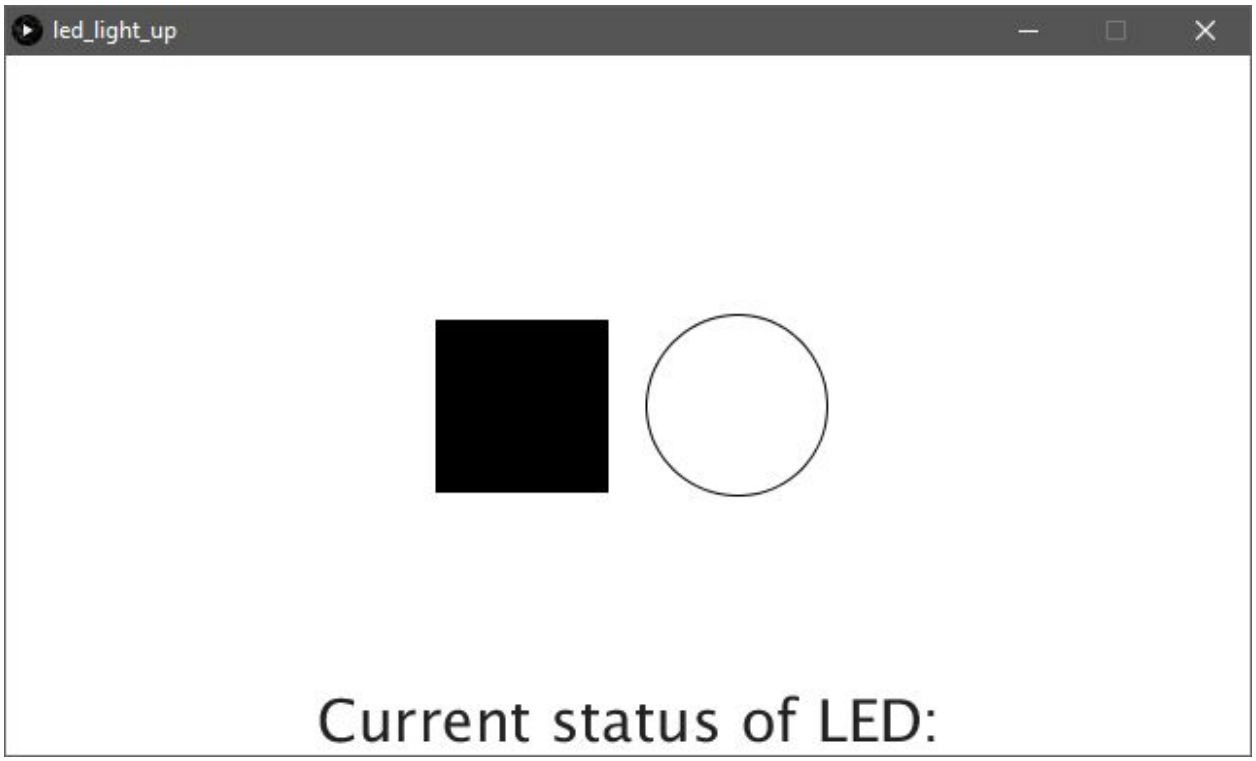
```
import processing.serial.*;

int rectX, rectY;     // Position of square button
int circleX, circleY;  // Position of circle button
int rectSize = 90;    // Diameter of rect
int circleSize = 93;   // Diameter of circle
color rectColor, circleColor, baseColor;
color rectHighlight, circleHighlight;
color currentColor;
boolean rectOver = false;
boolean circleOver = false;

String current_status = "0";
String myText = "Current status of LED:";
Serial myPort;  // Create object from Serial class
```

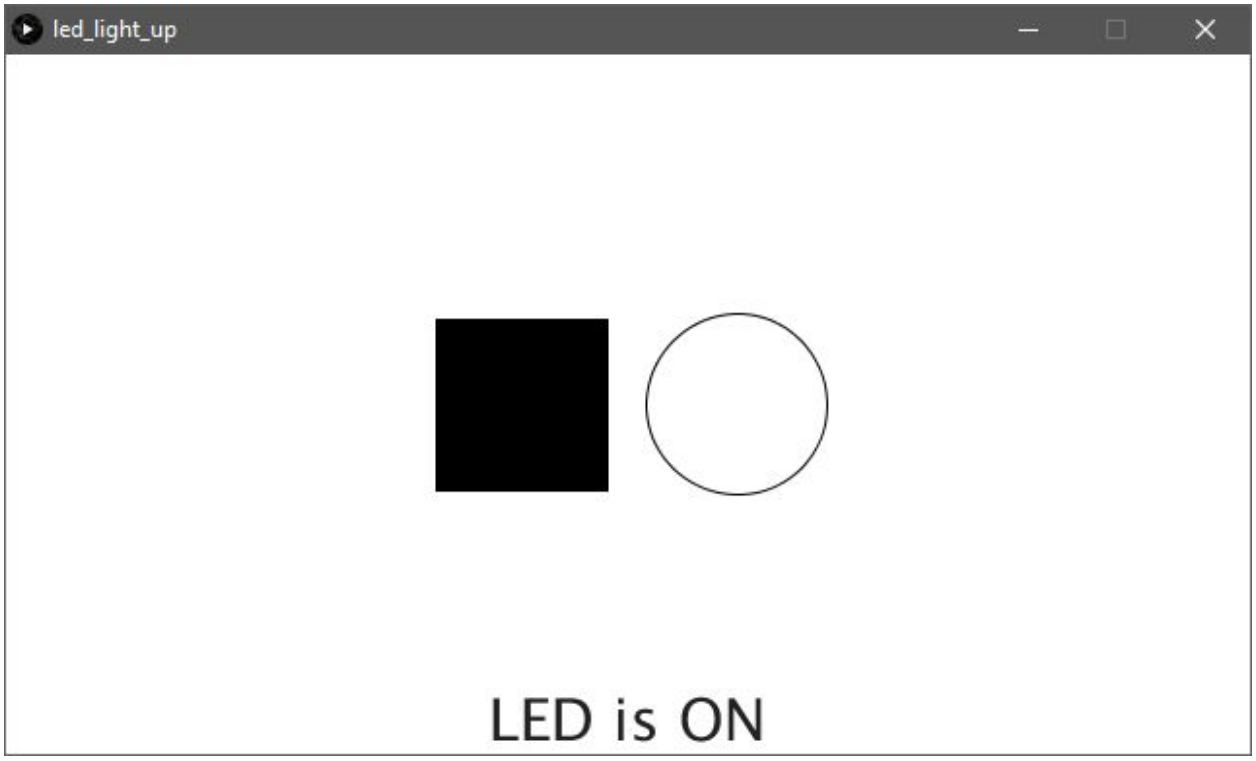
```
int val;        // Data received from the serial port

void setup() {
  size(640, 360);

  textAlign(CENTER, BOTTOM);
  textSize(30);
  fill(0);

  rectColor = color(0);
  rectHighlight = color(51);
  circleColor = color(255);
  circleHighlight = color(204);
  baseColor = color(102);
  currentColor = baseColor;
  circleX = width/2+circleSize/2+10;
  circleY = height/2;
  rectX = width/2-rectSize-10;
  rectY = height/2-rectSize/2;
  ellipseMode(CENTER);

  String portName = "COM3";
  myPort = new Serial(this, portName, 9600);
}

void draw() {
  background(255);
  fill(34);
  text(myText, 0, 0, width, height);

  update(mouseX, mouseY);


  if (rectOver) {
      fill(rectHighlight);
  } else {
      fill(rectColor);
  }
  stroke(255);
```

```
  rect(rectX, rectY, rectSize, rectSize);

  if (circleOver) {
      fill(circleHighlight);
  } else {
      fill(circleColor);
  }
  stroke(0);
  ellipse(circleX, circleY, circleSize, circleSize);
}

void update(int x, int y) {
  if ( overCircle(circleX, circleY, circleSize) ) {
      circleOver = true;
      rectOver = false;
  } else if ( overRect(rectX, rectY, rectSize, rectSize) ) {
      rectOver = true;
      circleOver = false;
  } else {
      circleOver = rectOver = false;
  }
}

void mousePressed() {
  if (circleOver) {
      myPort.write('H');
      fill(0);
      myText  = "LED is ON";
  }
  if (rectOver) {
      myPort.write('L');
      fill(0);
      myText  = "LED is OFF";
  }
}

boolean overRect(int x, int y, int width, int height)  {
  if (mouseX >= x && mouseX <= x+width &&
      mouseY >= y && mouseY <= y+height) {
```

```
      return true;
  } else {
      return false;
  }
}

boolean overCircle(int x, int y, int diameter) {
  float disX = x - mouseX;
  float disY = y - mouseY;
  if (sqrt(sq(disX) + sq(disY)) < diameter/2 ) {
      return true;
  } else {
      return false;
  }
}
```

Screenshots:




led_light_up
LED is ON