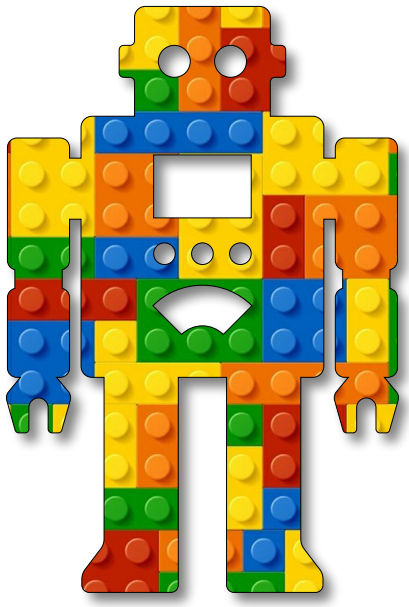
Intelligent Machines Module B Unit #4 RGB LED



Content

Module B Unit #4 RGB LED

Introduction to RGB LEDs

We are going to light up the RGB LED as the mouse passes over (hovers) over a circle with that colour in it. The `port.js` sketch remains the same.

Sketch B4.1 starting again

❗ The Arduino sketch

We will start with a basic sketch initialising the three RGB LED colours. Basically, removing everything inside the `if (Serial.available() > 0)` loop.

Arduino sketch

```
const int ledPinRed = LEDR;
const int ledPinGreen = LEDG;
const int ledPinBlue = LEDB;


void setup()
{
  pinMode(ledPinRed, OUTPUT);
  pinMode(ledPinBlue, OUTPUT);
  pinMode(ledPinGreen, OUTPUT);
  Serial.begin(9600);
}


void loop()
{
  if (Serial.available() > 0)
  {


  }
}
```

Notes

Setting up all three built-in RGB LED.

Sketch B4.2 command

We create the variable command (as a string variable). Read all the data coming in as a string until we get a new line. We will be sent a string from p5.js: RED, BLUE, or GREEN.

Arduino sketch

```
const int ledPinRed = LEDR;
const int ledPinGreen = LEDG;
const int ledPinBlue = LEDB;

void setup()
{
  pinMode(ledPinRed, OUTPUT);
  pinMode(ledPinBlue, OUTPUT);
  pinMode(ledPinGreen, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  if (Serial.available() > 0)
  {
    String command = Serial.readStringUntil('\n');
    command.trim();
  }
}
```

📝 Notes

Trim off any spaces.

🛠️ Code Explanation

`String command = Serial.readStringUntil('\n');`	Receives a word not a number
`command.trim();`	Removes any spaces

Sketch B4.3 first the red

If we receive the string RED (text), then we switch the red LED on.

Arduino

```
const int ledPinRed = LEDR;
const int ledPinGreen = LEDG;
const int ledPinBlue = LEDB;

void setup()
{
  pinMode(ledPinRed, OUTPUT);
  pinMode(ledPinBlue, OUTPUT);
  pinMode(ledPinGreen, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  if (Serial.available() > 0)
  {
    String command = Serial.readStringUntil('\n');
    command.trim();
    if (command == "RED")
    {
      digitalWrite(ledPinRed, LOW);
    }
  }
}
```

Notes

Remember that for the RGB LED, the logic is reversed: LOW is on! When it reads the word RED, it switches the LED on.

Sketch B4.4 and for the other colours

Same with the GREEN and BLUE.

Sketch

```
const int ledPinRed = LEDR;
const int ledPinGreen = LEDG;
const int ledPinBlue = LEDB;

void setup()
{
  pinMode(ledPinRed, OUTPUT);
  pinMode(ledPinBlue, OUTPUT);
  pinMode(ledPinGreen, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  if (Serial.available() > 0)
  {
    String command = Serial.readStringUntil('\n');
    command.trim();
    if (command == "RED")
    {
      digitalWrite(ledPinRed, LOW);
    }
    if (command == "GREEN")
    {
      digitalWrite(ledPinGreen, LOW);
    }
    if (command == "BLUE")
    {
      digitalWrite(ledPinBlue, LOW);
    }
  }
}
```

🗒️ Notes

The same for the other colours.

Sketch B4.5 everything off

If no string is sent, then we do not want any colour from the RGB LED on.

Sketch

```
const int ledPinRed = LEDR;
const int ledPinGreen = LEDG;
const int ledPinBlue = LEDB;

void setup()
{
  pinMode(ledPinRed, OUTPUT);
  pinMode(ledPinBlue, OUTPUT);
  pinMode(ledPinGreen, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  if (Serial.available() > 0)
  {
    String command = Serial.readStringUntil('\n');
    command.trim();
    if (command == "RED")
    {
      digitalWrite(ledPinRed, LOW);
    }
    if (command == "GREEN")
    {
      digitalWrite(ledPinGreen, LOW);
    }
    if (command == "BLUE")
    {
      digitalWrite(ledPinBlue, LOW);
    }
    else if (command == "HIGH")
    {
      digitalWrite(ledPinRed, HIGH);
```

```
      digitalWrite(ledPinGreen, HIGH);
      digitalWrite(ledPinBlue, HIGH);
    }
  }
}
```

🗒️ Notes

For the final condition, use the `else if()`. If you upload now, you might get all three `on` (white) at the same time.

Sketch B4.6 starting sketch

❗ The main sketch.js

We will draw three circles and colour them according to the RGB LEDs.

sketch.js

```
function setup()
{
  createCanvas(400, 400)
  navigation()
}

function draw()
{
  background(220)
  fill('red')
  circle(150, 100, 80)
  fill('green')
  circle(150, 200, 80)
  fill('blue')
  circle(150, 300, 80)
}
```

Notes

This should be fairly self-explanatory.

Figure B4.6


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Arduino p5js
sketch.js
Saved: 1 minute ago
Preview
1 function setup()
2 {
3 createCanvas(400, 400)
4 navigation()
5 }
6
7 function draw()
8 {
9 background(220)
10 fill('red')
11 circle(150, 100, 80)
12 fill('green')
13 circle(150, 200, 80)
14 fill('blue')
15 circle(150, 300, 80)
16 }
Console
Clear
port open

Sketch B4.7 mouse distance

We need to calculate the distance of the mouse to the centre of each circle.

sketch.js

```
let redDistance
let greenDistance
let blueDistance

function setup()
{
  createCanvas(400, 400)
  navigation()
}

function draw()
{
  background(220)
  fill('red')
  circle(150, 100, 80)
  fill('green')
  circle(150, 200, 80)
  fill('blue')
  circle(150, 300, 80)
  redDistance = dist(mouseX, mouseY, 150, 100)
  greenDistance = dist(mouseX, mouseY, 150, 200)
  blueDistance = dist(mouseX, mouseY, 150, 300)
}
```

Notes

Nothing new to see. The `dist()` function measures the distance between two sets of co-ordinates.

Sketch B4.8 check the distance

Now that we have calculated the distance between the mouse and the centre of each circle, we need to do something with that information. We will send a string when the mouse is inside the circle.

sketch.js

```
let redDistance
let greenDistance
let blueDistance

function setup()
{
  createCanvas(400, 400)
  navigation()
}

function draw()
{
  background(220)
  fill('red')
  circle(150, 100, 80)
  fill('green')
  circle(150, 200, 80)
  fill('blue')
  circle(150, 300, 80)
  redDistance = dist(mouseX, mouseY, 150, 100)
  greenDistance = dist(mouseX, mouseY, 150, 200)
  blueDistance = dist(mouseX, mouseY, 150, 300)
  if (redDistance <= 40)
  {
    serial.write("RED\n")
  }
}
```

Notes

If you run this sketch, you will see the red part of the LED come on once when the mouse is inside the circle; however, it stays on when the mouse moves outside the circle. We will fix that issue in due course.

Sketch B4.9 and the other colours

Now we will add the other colours. But all that we get is white as each colour adds to the mix.

sketch.js

```
let redDistance
let greenDistance
let blueDistance

function setup()
{
  createCanvas(400, 400)
  navigation()
}

function draw()
{
  background(220)
  fill('red')
  circle(150, 100, 80)
  fill('green')
  circle(150, 200, 80)
  fill('blue')
  circle(150, 300, 80)
  redDistance = dist(mouseX, mouseY, 150, 100)
  greenDistance = dist(mouseX, mouseY, 150, 200)
  blueDistance = dist(mouseX, mouseY, 150, 300)
  if (redDistance <= 40)
  {
    serial.write("RED\n")
  }
  else if (greenDistance <= 40)
  {
    serial.write("GREEN\n")
  }
  else if (blueDistance <= 40)
  {
```

```
    serial.write("BLUE\n")
  }
}
```

🗒️ Notes

The commands RED, GREEN, and BLUE are sent to the Arduino.

Sketch B4.10 the off switch

We need the RGB LED to reset, so that we can see the colours individually. Also, we need them to be off altogether when not hovering over a circle.

sketch.js

```
let redDistance
let greenDistance
let blueDistance

function setup()
{
  createCanvas(400, 400)
  navigation()
}

function draw()
{
  background(220)
  fill('red')
  circle(150, 100, 80)
  fill('green')
  circle(150, 200, 80)
  fill('blue')
  circle(150, 300, 80)
  redDistance = dist(mouseX, mouseY, 150, 100)
  greenDistance = dist(mouseX, mouseY, 150, 200)
  blueDistance = dist(mouseX, mouseY, 150, 300)
  if (redDistance <= 40)
  {
    serial.write("RED\n")
  }
  else if (greenDistance <= 40)
  {
    serial.write("GREEN\n")
  }
  else if (blueDistance <= 40)
  {
```

```
    serial.write("BLUE\n")
  }
  else
  {
    serial.write("HIGH\n")
  }
}
```

🗒️ Notes

Now it should work perfectly. The RGB LED should display the colour of the circle you hover over, and when you are not over any circle, then nothing should be on.

Sketch B4.11 a bit of a tweak

Just to finish off, let's add a circle that will display the colour of the RGB LED that is on. Just for good measure.

sketch.js

```
let redDistance
let greenDistance
let blueDistance

function setup()
{
  createCanvas(400, 400)
  navigation()
}

function draw()
{
  background(220)
  fill('red')
  circle(150, 100, 80)
  fill('green')
  circle(150, 200, 80)
  fill('blue')
  circle(150, 300, 80)
  redDistance = dist(mouseX, mouseY, 150, 100)
  greenDistance = dist(mouseX, mouseY, 150, 200)
  blueDistance = dist(mouseX, mouseY, 150, 300)
  if (redDistance <= 40)
  {
    serial.write("RED\n")
    fill('red')
  }
  else if (greenDistance <= 40)
  {
    serial.write("GREEN\n")
    fill('green')
  }
```

```
  else if (blueDistance <= 40)
  {
    serial.write("BLUE\n")
    fill('blue')
  }
  else
  {
    serial.write("HIGH\n")
    noFill()
  }
  circle(300, 200, 80)
}
```

Notes

Now we have an indicator.

Figure B4.11