

```

#include <ESP8266.h>
#include <BlynkSimpleShieldEsp8266.h>
#include "auth.h"
#define EspSerial Serial1 // Use Serial1 to communicate to the
ESP8266
ESP8266 wifi(EspSerial);
const char auth[] = BLYNK_TOKEN;

#include <Adafruit_NeoPixel.h>
#include <avr/power.h>

#define PIN1 4
#define PIN2 5
#define PIN3 6
#define PIN4 14
#define PIN5 15
#define PIN6 16
#define NUM_OF_PIXELS 12

int howBright = 25;
int howBrightBoth = 100;
int delaySteps = 50;
int r = 0;
int g = 0;
int b = 0;
int rBoth = 255;
int gBoth = 255;
int bBoth = 255;

Adafruit_NeoPixel ring1 = Adafruit_NeoPixel(NUM_OF_PIXELS, PIN1, NEO_GRB +
NEO_KHZ800);
Adafruit_NeoPixel ring2 = Adafruit_NeoPixel(NUM_OF_PIXELS, PIN2, NEO_GRB +
NEO_KHZ800);
Adafruit_NeoPixel ring3 = Adafruit_NeoPixel(NUM_OF_PIXELS, PIN3, NEO_GRB +
NEO_KHZ800);
Adafruit_NeoPixel ring4 = Adafruit_NeoPixel(NUM_OF_PIXELS, PIN4, NEO_GRB +
NEO_KHZ800);
Adafruit_NeoPixel ring5 = Adafruit_NeoPixel(NUM_OF_PIXELS, PIN5, NEO_GRB +
NEO_KHZ800);
Adafruit_NeoPixel ring6 = Adafruit_NeoPixel(NUM_OF_PIXELS, PIN6, NEO_GRB +
NEO_KHZ800);

const int sensorPin = A0;
const int ledPinCal = 9;

int pirPinR = 2;
int ledPinR = 7;
int pirPinL = 3;
int ledPinL = 8;

void setNeoPixelColorLeft(byte red, byte green, byte blue, byte brightness) {
  uint32_t color = ring1.Color((red * brightness) >> 8, (green * brightness)
>> 8, (blue * brightness) >> 8);
  for (int i = 0; i < NUM_OF_PIXELS; i++) {
    ring1.setPixelColor(i, color);
    ring2.setPixelColor(i, color);
    ring3.setPixelColor(i, color);
  }
}

```

```

    ring1.show();
    ring2.show();
    ring3.show();
    delay(delaySteps);
}
}

void setNeoPixelColorRight(byte red, byte green, byte blue, byte brightness)
{
    uint32_t color = ring4.Color((red * brightness) >> 8, (green * brightness)
>> 8, (blue * brightness) >> 8);
    for (int i = 0; i < NUM_OF_PIXELS; i++) {
        ring4.setPixelColor(i, color);
        ring5.setPixelColor(i, color);
        ring6.setPixelColor(i, color);
        ring4.show();
        ring5.show();
        ring6.show();
        delay(delaySteps);
    }
}

void initNeoPixels() {
    ring1.begin();
    ring2.begin();
    ring3.begin();
    ring4.begin();
    ring5.begin();
    ring6.begin();

    ring1.show();
    ring2.show();
    ring3.show();
    ring4.show();
    ring5.show();
    ring6.show();
}

BLYNK_WRITE(V1) { r = param.asInt(); }
BLYNK_WRITE(V2) { g = param.asInt(); }
BLYNK_WRITE(V3) { b = param.asInt(); }

BLYNK_READ(V4) { Blynk.virtualWrite(V4, analogRead(sensorPin)); }

BLYNK_WRITE(V5) { rBoth = param.asInt(); }
BLYNK_WRITE(V6) { gBoth = param.asInt(); }
BLYNK_WRITE(V7) { bBoth = param.asInt(); }

BLYNK_WRITE(V8) { howBright = param.asInt(); }
BLYNK_WRITE(V9) { howBrightBoth = param.asInt(); }

void setup() {

    initNeoPixels();

    EspSerial.begin(115200);
    Serial.begin(115200);
}

```

```

Blynk.begin(auth, wifi, WIFI_SSID, WIFI_PASSWORD);

pinMode(pirPinL, INPUT);
pinMode(ledPinL, OUTPUT);

pinMode(pirPinR, INPUT);
pinMode(ledPinR, OUTPUT);

howBright = 25;
howBrightBoth = 100;

r = 0;
g = 255;
b = 0;

rBoth = 255;
gBoth = 255;
bBoth = 255;
}

void activateLeftLedRings() {
  // digitalWrite(ledPinL, HIGH);
  setNeoPixelColorLeft(r, g, b, howBright);
}

void deactivateLeftLedRings() {
  // digitalWrite(ledPinL, LOW);
  setNeoPixelColorLeft(0, 0, 0, 0);
}

void activateRightLedRings() {
  // digitalWrite(ledPinR, HIGH);
  setNeoPixelColorRight(r, g, b, howBright);
}

void deactivateRightLedRings() {
  // digitalWrite(ledPinR, LOW);
  setNeoPixelColorRight(0, 0, 0, 0);
}

void activateBothLedRings() {
  // digitalWrite(ledPinL, HIGH);
  // digitalWrite(ledPinR, HIGH);
  setNeoPixelColorRight(rBoth, gBoth, bBoth, howBrightBoth);
  setNeoPixelColorLeft(rBoth, gBoth, bBoth, howBrightBoth);
}

bool isDark() {
  int sensorValue = getLightness();
  // analogWrite(ledPinCal, map(sensorValue, 0, 600, 0, 255));
  return sensorValue <= 5;
}

bool motionDetectedLeft() {

```

```

    return digitalRead(pirPinL) == HIGH;
}

bool motionDetectedRight() {
    return digitalRead(pirPinR) == HIGH;
}

int getLightness() {
    int sensorValue = analogRead(sensorPin);

    // enable this line to check the sensorvalue in the serial monitor
    int previousSensorValue = 0 ;
    // if(sensorValue != previousSensorValue){
    //     Serial.println(sensorValue); // Print serial line to see value
of the photocell sensor value
    //     previousSensorValue = sensorValue;
    // }
    return sensorValue;
}

bool leftLedRingsActive = false;
bool rightLedRingsActive = false;

void loop() {

Blynk.run();

    if (isDark()) {
        if (motionDetectedLeft() && !leftLedRingsActive) {
            Serial.println("Motion detected on left side of the bed, activating
left neo pixels");
            activateLeftLedRings();
            leftLedRingsActive = true;
        }

        if (motionDetectedRight() && !rightLedRingsActive) {
            Serial.println("Motion detected on the right side of the bed,
activating right neo pixels");
            activateRightLedRings();
            rightLedRingsActive = true;
        }

        if (rightLedRingsActive && leftLedRingsActive) {
            Serial.println("Full motion detected, activating all neo pixels");
            activateBothLedRings();
        }

    }

    if (!isDark()){
        deactivateRightLedRings();
        deactivateLeftLedRings();
        leftLedRingsActive = false;
        rightLedRingsActive = false;
    }

    if (!motionDetectedLeft() && leftLedRingsActive) {

```

```
    Serial.println("No more motion detected on left side of the bed,  
deactivating left neo pixels");  
    deactivateLeftLedRings();  
    leftLedRingsActive = false;  
}  
  
if (!motionDetectedRight() && rightLedRingsActive) {  
    Serial.println("No more motion detected on the right side of the bed,  
deactivating right neo pixels");  
    deactivateRightLedRings();  
    rightLedRingsActive = false;  
}  
  
}
```

You would also need to add the `auth.h` file to your Arduino project. This would look something like this:

```
#define BLYNK_TOKEN          "6f74r7a1735a4bd89f810accb8abbfca"  
#define WIFI_SSID           "YourNetworkNameGoesHere"  
#define WIFI_PASSWORD       "YourNetworkPasswordGoesHere"
```