int LDRValue = A0; // LDR sensor AO pin to A0 Connected

int SoundSensor=2; // Sound Sensor Digital Pin D0 connected to pin 2

int sleep = 4; //Sleep option switch assign to pin 4

int read = 5; //Read option switch assign to pin 5

int work = 6; //Work option switch assign to pin 6

int game = 7; //Game option switch assign to pin 7

int LED = 8; // This LED show the brightness Coz 4Pin RGB LED cant controle brightness

int R = 9; //add red controler to pin 9

int G = 10;//add green controler to pin 11

int B = 11;//add blue controler to pin 10

boolean RStatus=false;

boolean GStatus=false;

boolean BStatus=false;

void setup() {

pinMode(SoundSensor,INPUT);

pinMode(A0, INPUT);

pinMode(R,OUTPUT);

pinMode(G,OUTPUT);

pinMode(B,OUTPUT);

Serial.begin(9600); //initialize serial

}

void loop() {

int SoundData=digitalRead(SoundSensor); // take readings from sound sensor

Serial.println(SoundData);//print the value

int sleepMod = digitalRead(sleep);//take readings from mod switch

int readpMod = digitalRead(read);//take readings from mod switch

int workMod = digitalRead(work);//take readings from mod switch

int gameMod = digitalRead(game);//take readings from mod switch

if(SoundData==1){//if detect a clap

if(RStatus==false){//if lights off

RStatus=true;

if (sleepMod == HIGH){//check the Mod Sleep

LDRValue = analogRead(A0); // read the value from the LDR sensor

Serial.println(LDRValue); // print the LDR sensor reading so you know its range

int Bright = 0 ; // set brightness to 0

//216- Normal Light brightness (216\*3.7)

Bright = 216-(LDRValue/3.7) ; //Calculate needed brightness to room

if (Bright < 0){ //if room have enough brightness lights off

digitalWrite(R,LOW);

digitalWrite(G,LOW);

digitalWrite(B,LOW);

}

else if (Bright > 0){//check the Mod Sleep

analogWrite(R, 255); // Red colour is better for sleeping

analogWrite(G, 0);

analogWrite(B, 0);

delay(10); // Delay a little bit to improve simulation performance

}

delay(100); // Wait for 100 millisecond(s)

}

else if (readpMod == HIGH){//check the Mod Read

LDRValue = analogRead(A0); // read the value from the LDR sensor

Serial.println(LDRValue); // print the LDR sensor reading so you know its range

int Bright = 0 ; // set brightness to 0

//216- Normal Light brightness (216\*3.7)

Bright = 216-(LDRValue/3.7) ; //Calculate needed brightness to room

if (Bright < 0){ //if room have enough brightness lights off

digitalWrite(R,LOW);

digitalWrite(G,LOW);

digitalWrite(B,LOW);

}

else if (Bright > 0){//if room is dark

analogWrite(R, 255); // Yellow colour is better for Reading

analogWrite(G, 255);

analogWrite(B, 0);

delay(10); // Delay a little bit to improve simulation performance

}

delay(100); // Wait for 100 millisecond(s)

}

else if (workMod == HIGH){//check the Mod Work

LDRValue = analogRead(A0); // read the value from the LDR sensor

Serial.println(LDRValue); // print the LDR sensor reading so you know its range

int Bright = 0 ; // set brightness to 0

//216- Normal Light brightness (216\*3.7)

Bright = 216-(LDRValue/3.7) ; //Calculate needed brightness to room

if (Bright < 0){ //if room have enough brightness lights off

digitalWrite(R,LOW);

digitalWrite(G,LOW);

digitalWrite(B,LOW);

}

else if (Bright > 0){//if room is dark

analogWrite(R, 255); // White colour is better for Working

analogWrite(G, 255);

analogWrite(B, 255);

delay(10); // Delay a little bit to improve simulation performance

}

delay(100); // Wait for 100 millisecond(s)

}

else if (gameMod == HIGH){//check the Mod Game

LDRValue = analogRead(A0); // read the value from the LDR sensor

Serial.println(LDRValue); // print the LDR sensor reading so you know its range

int Bright = 0 ; // set brightness to 0

//216- Normal Light brightness (216\*3.7)

Bright = 216-(LDRValue/3.7) ; //Calculate needed brightness to room

if (Bright < 0){ //if room have enough brightness lights off

digitalWrite(R,LOW);

digitalWrite(G,LOW);

digitalWrite(B,LOW);

}

else if (Bright > 0){//if room is dark

analogWrite(R, 255); // Blue colour is better for Gaming

analogWrite(G, 0);

analogWrite(B, 255);

delay(10); // Delay a little bit to improve simulation performance

}

delay(100); // Wait for 100 millisecond(s)

}

}

else if(RStatus==true){

RStatus=false;

digitalWrite(R,LOW);

digitalWrite(G,LOW);

digitalWrite(B,LOW);

}}}