

```

// Pacman/Ghost LED Matrix

// Monserrath Velasco
// Santiago Guerra

// Import of LED strip library
#include <FastLED.h>

//Import of temperature sensor library
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>

//setting up the LED strip
#define NUM_LEDS 71
#define DATA_PIN 7

//setting up the LED strip's brightness
#define BRIGHTNESS 100

CRGB leds[NUM_LEDS]; //setting up the block of memory used for LED data

int buttonPin = 4; //assigning the push button pin
int buttonState = 0; // variable for reading the pushbutton status
int ledflag = 0; // LED status flag
int sensor = A4; //assigning the push button pin
Adafruit_BME280 bme; //creating an instance of the temperature sensor

void setup() {
  FastLED.addLeds<NEOPIXEL, DATA_PIN>(leds, NUM_LEDS); //defining the array of LEDs
  FastLED.setBrightness(BRIGHTNESS); //setting the LEDs brightness
  Serial.begin(9600); //the bound rate for serial monitor

  pinMode(buttonPin, INPUT); // defining pushbutton as an input

  if (!bme.begin(0x76)) {
    Serial.println("Could not find temperature");
    while(1);
  }
}

//Function to reset (turn off) all the LED strip
void reset() {
  for (int dot = 0; dot < NUM_LEDS; dot++) {
    leds[dot] = CRGB::Black; //turn off each LED
  }
}

```

```

    FastLED.show();
    delay(20);
  }
}

//Function to display the ghost LED animation
void ghost_loop() {

  //creating the initial "body"
  for (int dot = 0; dot < 6; dot++) {
    leds[dot] = CRGB::Red;
    FastLED.show();
    delay(200);
  }

  delay(200);

  for (int dot = 7; dot < 11; dot++) {
    leds[dot] = CRGB::Red;
    FastLED.show();
    delay(200);
  }

  delay(200);

  //creating the ghost's eyes
  leds[11] = CRGB::Blue;
  FastLED.show();
  delay(200);
  leds[12] = CRGB::White;
  FastLED.show();
  delay(200);
  leds[13] = CRGB::Red;
  FastLED.show();
  delay(200);
  leds[14] = CRGB::Red;
  FastLED.show();
  delay(200);
  leds[15] = CRGB::White;
  FastLED.show();
  delay(200);
  leds[16] = CRGB::White;
  FastLED.show();

```

```
delay(200);
```

```
//continuing with the ghost's "body"  
for (int dot = 17; dot < 22 ; dot++) {  
  leds[dot] = CRGB::Red;  
  FastLED.show();  
  delay(200);  
}
```

```
delay(200);
```

```
//continuing with the ghost's "body"  
for (int dot = 23; dot < 40 ; dot++) {  
  leds[dot] = CRGB::Red;  
  FastLED.show();  
  delay(200);  
}
```

```
delay(200);
```

```
//continuing with the ghost's "body"  
for (int dot = 41; dot < 45 ; dot++) {  
  leds[dot] = CRGB::Red;  
  FastLED.show();  
  delay(200);  
}
```

```
delay(200);
```

```
//continuing with the ghost's eyes  
leds[45] = CRGB::Blue;  
FastLED.show();  
delay(200);  
leds[46] = CRGB::White;  
FastLED.show();  
delay(200);  
leds[47] = CRGB::Red;  
FastLED.show();  
delay(200);  
leds[48] = CRGB::Red;  
FastLED.show();  
delay(200);  
leds[49] = CRGB::Red;  
FastLED.show();
```

```
delay(200);
leds[50] = CRGB::White;
FastLED.show();
delay(200);
leds[51] = CRGB::White;
FastLED.show();

delay(200);

//finishing with the ghost's "body"
for (int dot = 52; dot < 57 ; dot++) {
  leds[dot] = CRGB::Red;
  FastLED.show();
  delay(200);
}

delay(200);

//finishing with the ghost's "body"
for (int dot = 58; dot < NUM_LEDS ; dot++) {
  leds[dot] = CRGB::Red;
  FastLED.show();
  delay(200);
}

delay(200);

//moving ghost's "feet" by alternating colors
for (int i = 0; i < 5; i++) {
  leds[6] = CRGB::Red;
  leds[22] = CRGB::Red;
  leds[40] = CRGB::Red;
  leds[57] = CRGB::Red;
  FastLED.show();

  leds[5] = CRGB::Black;
  leds[21] = CRGB::Black;
  leds[39] = CRGB::Black;
  leds[56] = CRGB::Black;
  leds[70] = CRGB::Black;
  FastLED.show();

  delay(200);
```

```

    leds[6] = CRGB::Black;
    leds[22] = CRGB::Black;
    leds[40] = CRGB::Black;
    leds[57] = CRGB::Black;
    FastLED.show();

    leds[5] = CRGB::Red;
    leds[21] = CRGB::Red;
    leds[39] = CRGB::Red;
    leds[56] = CRGB::Red;
    leds[70] = CRGB::Red;
    FastLED.show();

    delay(200);

    //moving ghost's eyes by alternating colors
    leds[11] = CRGB::White;
    leds[16] = CRGB::Blue;
    FastLED.show();

    leds[45] = CRGB::White;
    leds[51] = CRGB::Blue;
    FastLED.show();

    delay(200);

    leds[11] = CRGB::Blue;
    leds[16] = CRGB::White;
    FastLED.show();

    leds[45] = CRGB::Blue;
    leds[51] = CRGB::White;
    FastLED.show();
}
reset(); //turning off all the LEDs
}

//Function to display the ghost now turned blue (cold temperature)
void ghost_loop_aqua() {

    //creating the ghost body (now Aqua)
    for (int dot = 0; dot < 6; dot++) {
        leds[dot] = CRGB::Aqua;
        FastLED.show();
    }
}

```

```
    delay(200);  
}
```

```
delay(200);
```

```
//creating the ghost body (now Aqua)  
for (int dot = 7; dot < 11; dot++) {  
    leds[dot] = CRGB::Aqua;  
    FastLED.show();  
    delay(200);  
}
```

```
delay(200);
```

```
//creating the ghost eyes  
leds[11] = CRGB::Blue;  
FastLED.show();  
delay(200);  
leds[12] = CRGB::White;  
FastLED.show();  
delay(200);  
leds[13] = CRGB::Aqua;  
FastLED.show();  
delay(200);  
leds[14] = CRGB::Aqua;  
FastLED.show();  
delay(200);  
leds[15] = CRGB::White;  
FastLED.show();  
delay(200);  
leds[16] = CRGB::White;  
FastLED.show();
```

```
delay(200);
```

```
//creating the ghost body (now Aqua)  
for (int dot = 17; dot < 22 ; dot++) {  
    leds[dot] = CRGB::Aqua;  
    FastLED.show();  
    delay(200);  
}
```

```
delay(200);
```

```
//creating the ghost body (now Aqua)
for (int dot = 23; dot < 40 ; dot++) {
  leds[dot] = CRGB::Aqua;
  FastLED.show();
  delay(200);
}
```

```
delay(200);
```

```
//creating the ghost body (now Aqua)
for (int dot = 41; dot < 45 ; dot++) {
  leds[dot] = CRGB::Aqua;
  FastLED.show();
  delay(200);
}
```

```
delay(200);
```

```
//creating the ghost's eyes
leds[45] = CRGB::Blue;
FastLED.show();
delay(200);
leds[46] = CRGB::White;
FastLED.show();
delay(200);
leds[47] = CRGB::Aqua;
FastLED.show();
delay(200);
leds[48] = CRGB::Aqua;
FastLED.show();
delay(200);
leds[49] = CRGB::Aqua;
FastLED.show();
delay(200);
leds[50] = CRGB::White;
FastLED.show();
delay(200);
leds[51] = CRGB::White;
FastLED.show();
```

```
delay(200);
```

```
//finishing the ghost body (now Aqua)
```

```

for (int dot = 52; dot < 57 ; dot++) {
  leds[dot] = CRGB::Aqua;
  FastLED.show();
  delay(200);
}

delay(200);

//finishing the ghost body (now Aqua)
for (int dot = 58; dot < NUM_LEDS ; dot++) {
  leds[dot] = CRGB::Aqua;
  FastLED.show();
  delay(200);
}

delay(200);

for (int i = 0; i < 5; i++) {
  //moving ghost's "feet" by alternating colors
  leds[6] = CRGB::Aqua;
  leds[22] = CRGB::Aqua;
  leds[40] = CRGB::Aqua;
  leds[57] = CRGB::Aqua;
  FastLED.show();

  leds[5] = CRGB::Black;
  leds[21] = CRGB::Black;
  leds[39] = CRGB::Black;
  leds[56] = CRGB::Black;
  leds[70] = CRGB::Black;
  FastLED.show();

  delay(200);

  leds[6] = CRGB::Black;
  leds[22] = CRGB::Black;
  leds[40] = CRGB::Black;
  leds[57] = CRGB::Black;
  FastLED.show();

  leds[5] = CRGB::Aqua;
  leds[21] = CRGB::Aqua;
  leds[39] = CRGB::Aqua;

```



```

    leds[56] = CRGB::Aqua;
    leds[70] = CRGB::Aqua;
    FastLED.show();

    delay(200);

    //moving ghost's eyes by alternating colors
    leds[11] = CRGB::White;
    leds[16] = CRGB::Blue;
    FastLED.show();

    leds[45] = CRGB::White;
    leds[51] = CRGB::Blue;
    FastLED.show();

    delay(200);

    leds[11] = CRGB::Blue;
    leds[16] = CRGB::White;
    FastLED.show();

    leds[45] = CRGB::Blue;
    leds[51] = CRGB::White;
    FastLED.show();

}
    reset(); //turning off all the LEDs
}

//Function to display the Pacman animation
void pacman_loop() {

    //creating the Pacman "body"
    for (int dot = 0; dot < 3; dot++) {
        leds[dot] = CRGB::Yellow;
        FastLED.show();
        delay(200);
    }

    delay(200);

    //creating the Pacman "body"
    for (int dot = 7; dot < 21; dot++) {

```

```
    leds[dot] = CRGB::Yellow;
    FastLED.show();
    delay(200);
}

delay(200);

//creating the Pacman "body"
for (int dot = 22; dot < 35; dot++) {
    leds[dot] = CRGB::Yellow;
    FastLED.show();
    delay(200);
}

delay(200);

//creating the Pacman "body"
for (int dot = 36; dot < 43; dot++) {
    leds[dot] = CRGB::Yellow;
    FastLED.show();
    delay(200);
}

delay(200);

//creating the Pacman "body"
for (int dot = 46; dot < 50; dot++) {
    leds[dot] = CRGB::Yellow;
    FastLED.show();
    delay(200);
}

delay(200);

leds[55] = CRGB::Yellow;
FastLED.show();

delay(200);

//loop to open/close Pacman's mouth
for (int i = 0; i < 5; i++) {
    //opening the Pacman's mouth
    leds[35] = CRGB::Yellow;
    leds[43] = CRGB::Yellow;
```

```
leds[45] = CRGB::Yellow;  
leds[50] = CRGB::Yellow;  
leds[54] = CRGB::Yellow;  
leds[58] = CRGB::Yellow;  
leds[64] = CRGB::Yellow;  
FastLED.show();
```

```
delay(300);
```

```
leds[44] = CRGB::Yellow;  
leds[51] = CRGB::Yellow;  
leds[53] = CRGB::Yellow;  
leds[59] = CRGB::Yellow;  
leds[63] = CRGB::Yellow;
```

```
FastLED.show();
```

```
delay(300);
```

```
leds[52] = CRGB::Yellow;  
leds[60] = CRGB::Yellow;  
leds[62] = CRGB::Yellow;  
leds[65] = CRGB::Yellow;  
leds[67] = CRGB::Yellow;  
FastLED.show();
```

```
delay(300);
```

```
leds[61] = CRGB::Yellow;  
leds[66] = CRGB::Yellow;  
FastLED.show();
```

```
delay(1500);
```

```
//closing the Pacman's mouth by turning off some LEDs
```

```
leds[66] = CRGB::Black;  
leds[61] = CRGB::Black;  
FastLED.show();
```

```
delay(300);
```

```
leds[52] = CRGB::Black;  
leds[60] = CRGB::Black;  
leds[62] = CRGB::Black;
```

```
leds[65] = CRGB::Black;
leds[67] = CRGB::Black;
FastLED.show();
```

```
delay(300);
```

```
leds[44] = CRGB::Black;
leds[51] = CRGB::Black;
leds[53] = CRGB::Black;
leds[59] = CRGB::Black;
leds[63] = CRGB::Black;
FastLED.show();
```

```
delay(300);
```

```
leds[35] = CRGB::Black;
leds[43] = CRGB::Black;
leds[45] = CRGB::Black;
leds[50] = CRGB::Black;
leds[54] = CRGB::Black;
leds[58] = CRGB::Black;
leds[64] = CRGB::Black;
FastLED.show();
```

```
}
```

```
delay(200);
```

```
//opening the Pacman's mouth
```

```
leds[35] = CRGB::Yellow;
leds[43] = CRGB::Yellow;
leds[45] = CRGB::Yellow;
leds[50] = CRGB::Yellow;
leds[54] = CRGB::Yellow;
leds[58] = CRGB::Yellow;
leds[64] = CRGB::Yellow;
FastLED.show();
```

```
delay(300);
```

```
leds[44] = CRGB::Yellow;
leds[51] = CRGB::Yellow;
leds[53] = CRGB::Yellow;
leds[59] = CRGB::Yellow;
```

```

    leds[63] = CRGB::Yellow;

    FastLED.show();

    delay(300);

    leds[52] = CRGB::Yellow;
    leds[60] = CRGB::Yellow;
    leds[62] = CRGB::Yellow;
    leds[65] = CRGB::Yellow;
    leds[67] = CRGB::Yellow;
    FastLED.show();

    delay(300);

    leds[61] = CRGB::Yellow;
    leds[66] = CRGB::Yellow;
    FastLED.show();

    //turning off all the LEDs
    reset();
    delay(1000);
}

void loop() {

    Serial.println("Temperature =");
    Serial.println(bme.readTemperature()); //displaying the captured temperature in the serial
    monitor

    buttonState = digitalRead(buttonPin); // reading the state of the pushbutton value

    if (buttonState == HIGH) { // checking if the pushbutton is pressed.
        if(ledflag==0 && bme.readTemperature() >= 28) { // check if the temperature is below 30
        degrees
            ledflag=1; // making status flag HIGH
            ghost_loop(); //calling the ghost animation
        }

        else if(ledflag==0 && bme.readTemperature() < 28) { // check if the temperature is below 23
        degrees
            ledflag=1; // making status flag HIGH
            ghost_loop_aqua(); //calling the ghost animation (cold temperature)
        }
    }
}

```

```
else if(ledflag==1) {  
    ledflag=0; // making status flag LOW  
    pacman_loop(); //calling the Pacman animation  
}  
}  
  
}
```