

```
// This #include statement was automatically added by the Particle IDE.

#include <neopixel.h>

#include "application.h"

SYSTEM_MODE(AUTOMATIC);

// IMPORTANT: Set pixel COUNT, PIN and TYPE

#define PIXEL_PIN D6 //sets the neopixels to digital pin 6

#define PIXEL_COUNT 33

#define PIXEL_TYPE WS2812B

Adafruit_NeoPixel strip = Adafruit_NeoPixel(PIXEL_COUNT, PIXEL_PIN, PIXEL_TYPE);

int fabrikenparticleweather = 0; //sets the weather event data to zero (nothing) to start with

int mist = D2; //sets the mister to digital pin 2

int pump = D3; //sets the pump to digital pin 3

int wind = D4; //sets the fan to digital pin 4

void setup()

{

  strip.begin();

  strip.show(); // Initialize all pixels to 'off'

  pinMode(pump, OUTPUT); //defines the pump as an output

  pinMode(mist, OUTPUT); //defines the mister as an output

  pinMode(wind, OUTPUT); //defines the fan as an output
```

```
digitalWrite(wind, LOW);
```

```
//sets the fan to of
```

```
digitalWrite(pump, LOW); //sets the pump to of
```

```
digitalWrite(mist, LOW); //sets the mister to of
```

```
//starts an event subscription for a event named " _____ " choose your own event name.
```

```
//Remember to change "fabrikensparticleweather" to whatever name you choose throughout the program!!!
```

```
//myHandler is the data the event sends, in our case numbers corresponding to the weather
```

```
//0 = nothing
```

```
//1 = clear weather
```

```
//2 = rain
```

```
//3 = cloudy weather
```

```
//4 = snow
```

```
//5 = partytime!
```

```
Particle.subscribe("fabrikensparticleweather", myHandler); //Remember to change "fabrikensparticleweather" to whatever name you choose throughout the program!!!
```

```
}
```

```
void loop() //tells the program what to do
```

```
{
```

```
if(fabrikensparticleweather == 1) {
```

```
clear(strip.Color(255, 255, 255), 150); // fills the leds with one colour variables are (Red,Green,Blue) speed) turns of mist and rain
```

```
digitalWrite(pump, LOW);
```

```
digitalWrite(mist, LOW);
```

```
    digitalWrite(wind, LOW);
}
else if(fabrikensparticleweather == 2) { //RAIN

    rain(100,10,20,50); ///sparkle a colour on the led strip and turns on the mist and rain

        digitalWrite(pump, HIGH);

        digitalWrite(mist, HIGH);

        digitalWrite(wind, HIGH);
    }
else if(fabrikensparticleweather == 3) { //CLOUDY

    clear(strip.Color(30, 20, 20), 1500); // fills the leds with one colour, variables are (Red,Green,Blue)
    speed) turns of rain and turns on mist

    digitalWrite(pump, LOW);

        digitalWrite(mist, HIGH);

        digitalWrite(wind, LOW);
    }
else if(fabrikensparticleweather == 4) { //SNOWY

    snow(100,10,20,50); ///sparkle a colour on the led strip and turns off the mist and rain

    digitalWrite(pump, LOW);

        digitalWrite(mist, LOW);

        digitalWrite(wind, LOW);
    }
else if(fabrikensparticleweather == 5) { //PARTY

    sparkleRainbow(100, 50, 100); ///sparkle a colour on the led strip and turns off the mist and rain

        digitalWrite(pump, HIGH);
```

```
    digitalWrite(mist, HIGH);
    digitalWrite(wind, HIGH);
}
else {
    clear();
}
}
```

```
void myHandler(const char *event, const char *data) {
    fabrikensparticleweather = atoi(data);
}
```

// Functions for Clear weather, sets the LEDs to white light and turns of the pump for rain, the mister for clouds and the fan for wind.

```
void clear(uint32_t c, uint8_t wait) {
    for(uint16_t i=0; i<strip.numPixels(); i++) {
        strip.setPixelColor(i, c);
        strip.show();
        delay(wait);
    }
}
```

// Functions for snowy weather, sets the LEDs to a choosen colour for rani.

```
void rain(uint16_t cycles, uint16_t wait, uint8_t density, uint32_t color) {

    const int numPixels = strip.numPixels();
```

```

    for(int i=0;i<cycles*numPixels;i++){
        for(int j=0;j<numPixels;j++){
            if((rand() % 1000) < density){
                strip.setPixelColor(j, 0,100,255);// Sets the
RED, GREEN, BLUE colour on the Neopixels
            }else{
                strip.setPixelColor(j,
dimColor(strip.getPixelColor(j),0.8));
            }
        }
        strip.show();
        delay(wait);
    }
}

```

// Functions for snowy weather, sets the LEDs to a choosen colour for snow.

```

void snow(uint16_t cycles, uint16_t wait, uint8_t density, uint32_t color) {

```

```

    const int numPixels = strip.numPixels();

```

```

        for(int i=0;i<cycles*numPixels;i++){
            for(int j=0;j<numPixels;j++){
                if((rand() % 1000) < density){
                    strip.setPixelColor(j, 100,100,255);// Sets
the RED, GREEN, BLUE colour on the Neopixels
                }else{

```

```
strip.setPixelColor(j,
dimColor(strip.getPixelColor(j),0.8));
    }
    }
    strip.show();
    delay(wait);
}
}
```

```
void fill() {
    uint16_t i;

    for(i=0; i<12; i++) {
        strip.setPixelColor(i, strip.Color(255, 0, 0));
    }
    strip.show();
}
```

```
void clear() {
    uint16_t i;

    for(i=0; i<12; i++) {
        strip.setPixelColor(i, strip.Color(0, 0, 0));
    }
    strip.show();
}
```

```
void pulse(uint8_t wait) {  
    uint16_t i, j;  
    for(j=0; j<256; j++) {  
        for(i=0; i<12; i++) {  
            strip.setPixelColor(i, strip.Color(j, 0, 0));  
        }  
        strip.show();  
        delay(wait);  
    }  
    for(j=255; j>0; j--) {  
        for(i=0; i<12; i++) {  
            strip.setPixelColor(i, strip.Color(j, 0, 0));  
        }  
        strip.show();  
        digitalWrite(pump, LOW);  
        delay(wait);  
    }  
}
```

```
void blink(uint16_t wait) {  
    uint16_t i, j;  
  
    for(i=0; i<12; i++) {  
        strip.setPixelColor(i, strip.Color(0, 0, 0));  
    }
```

```

}

strip.show();

delay(wait);

for(i=0; i<12; i++) {

    strip.setPixelColor(i, strip.Color(255, 0, 0));

}

strip.show();

delay(wait);

}

// new code from here

// Slightly different, this makes the rainbow equally distributed throughout
void rainbowPattern(uint8_t wait, uint8_t density, uint8_t cycles, float brightness) {

    uint16_t i, j;

    for(j=0; j<256*cycles; j++) { // 5 cycles of all colors on wheel

        for(i=0; i< strip.numPixels(); i++) {

            //strip.setPixelColor(i, Wheel(((i * (256 * density) / strip.numPixels() ) + j) & 255));

            strip.setPixelColor(i, dimColor(Wheel(((i * (200 * density) / strip.numPixels() ) + j) &
255),brightness));

            //1ms, 10 cycles, mit dimColor: 7.8sec

            //1ms, 10 cycles, one dimColor: 5.7sec

        }

        strip.show();

        delay(wait);

    }
}

```



```
}
```

```
void sparkleRainbow(uint16_t cycles, uint16_t wait, uint8_t density) {
```

```
    const int numPixels = strip.numPixels();
```

```
        for(int i=0;i<cycles*numPixels;i++){
```

```
            for(int j=0;j<numPixels;j++){
```

```
                if((rand() % 5000) < density){
```

```
                    strip.setPixelColor(j, Wheel((rand() %  
255)));
```

```
                }else{
```

```
                    strip.setPixelColor(j,  
dimColor(strip.getPixelColor(j),0.8));
```

```
                }
```

```
            }
```

```
        strip.show();
```

```
        delay(wait);
```

```
    }
```

```
}
```

```
uint32_t Wheel(byte WheelPos) {
```

```
    WheelPos = 255 - WheelPos;
```

```
    if(WheelPos < 85) {
```

```
        return strip.Color(255 - WheelPos * 3, 0, WheelPos * 3);
```

```
} else if(WheelPos < 170) {  
    WheelPos -= 85;  
    return strip.Color(0, WheelPos * 3, 255 - WheelPos * 3);  
} else {  
    WheelPos -= 170;  
    return strip.Color(WheelPos * 3, 255 - WheelPos * 3, 0);  
}  
}
```

```
uint32_t dimColor(uint32_t color, float amount) {  
    return  
        ((int)((color&0xFF0000) * amount) &0xFF0000) +  
        ((int)((color&0x00FF00) * amount) &0x00FF00) +  
        ((int)((color&0x0000FF) * amount) &0x0000FF);  
}
```