

BBC micro:bit

Bit:Commander Neopixels

Introduction

The Bit:Commander has 6 Neopixels. They are at the top of the board and handily labelled 0 to 5. The Neopixels are connected to pin 13 on the micro:bit.

The Neopixels on the Bit:Commander don't need any special treatment. They work with all of your standard Neopixel code. Be warned that these LEDs are bright. That's great when you are using the board from distance as a lighting effect. If you also want to be able to see the buttons when you press them, you will need to tone down the brightness a lot.



Programming

You program for the Neopixels as you would normally do. The following program cycles through the button colours, blinking them on and off.

```
from microbit import *
import neopixel

# Initialise neopixels
npix = neopixel.NeoPixel(pin13, 6)

# Light all neopixels with given colour
def light_all(col):
    for pix in range(0, len(npix)):
        npix[pix] = col
        npix.show()

red = (64,0,0)
green = (0,64,0)
blue = (0,0,64)
yellow = (64,24,0)
off = (0,0,0)

cols = [red,green,blue,yellow,off]

while True:
    for c in cols:
        light_all(c)
        sleep(1000)
        light_all(off)
        sleep(1000)
```

Simple Joystick/Button Example

This second program is a little cuter. You hold down one of the 4 pushbuttons and then move the joystick left or right. The button colour is 'wiped' across the LEDs in the direction indicated. Whilst it might not be the most useful of gadgets, it is quite a nice feeling to control the colours like this.

BBC Microbit

Collapse All

Expand All

+ Block Editor - The Basics

+ Block Editor - Components

+ Kodu - micro:bit Worlds

+ JavaScript Blocks

+ JavaScript Blocks - Exercises

+ Blocks - Bit:Bot

+ Blocks - Bit:Commander

+ MicroPython - Starting Off

+ MicroPython - Examples

+ MicroPython - Components

+ MicroPython - Breakout Boards

+ MicroPython - Exercises

+ MicroPython - Pi Accessories

+ MicroPython - Bit:Bot

- MicroPython - Bit:Commander

+ Bit:Commander

+ The Joystick

+ The Neopixels

+ The Potentiometer

+ The Pushbuttons

+ The Buzzer

+ Evasion Game

+ Light's Out Game

+ Simon Game

+ Bit:Bot/Robot Controller

+ Text Entry

+ Unicorn Commander

+ MicroPython - Projects

+ MicroPython - Visual Basic

+ Other - Odds & Ends



```
# Hold down a button and move the joystick left or right to  
# wipe a colour across the LED chain in the direction of  
# choice
```

```
from microbit import *  
import neopixel
```

```
# Initialise neopixels  
npix = neopixel.NeoPixel(pin13, 6)
```

```
red = (64,0,0)  
green = (0,64,0)  
blue = (0,0,64)  
yellow = (64,24,0)  
off = (0,0,0)  
button_pins = [pin12, pin15, pin14, pin16]  
ledcols = [red,blue,green,yellow]
```

```
def wipe(col, delay, left):  
    for pix in range(0, 6):  
        if left:  
            npix[pix] = col  
        else:  
            npix[5-pix] = col  
        npix.show()  
        sleep(delay)
```

```
while True:  
    # check button presses  
    # no multitouch, priority is y,g,b,r  
    chosen = off  
    for i in range(4):  
        if button_pins[i].read_digital():  
            chosen = ledcols[i]  
    # now check for joystickX for wipe direction  
    x = pin1.read_analog()  
    if x<150:  
        wipe(chosen,50,False)  
        sleep(500)  
    elif x>750:  
        wipe(chosen,60,True)  
        sleep(500)  
    sleep(50)
```

Exploring Further

There are a few other pages with Neopixel code on the site. Some of the functions there are usable here in a drop-in fashion.

One of the most obvious things to do is to use the LEDs as indicators, that a button is pressed, a trigger reading value reached or that a radio message has arrived. Most of your fancy electronic gadgets have sound and light indicators to make the experience better. Adding those features to your circuits and projects makes them seem more polished.