11/17/2018 multiwingspan

## MultiWingSpan

Home Programming Web Design Computer Science Twisting Puzzles

# BBC micro:bit LEDs With MicroPython

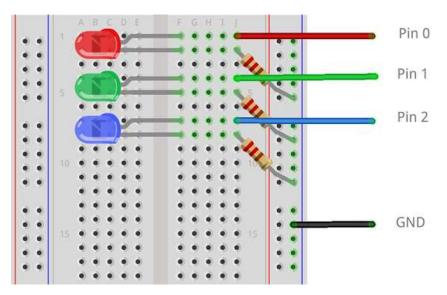
#### Introduction

You can use the external pins to connect more LEDs and have colours other than red. Using some LEDs gives us a chance to use the 3 main large pins as digital and analogue outputs.

This project uses 3 LEDs. Each LED will need a resistor. Here I've used 220 Ohm resistors, one for each of the LEDs used.

#### **Making The Circuit**

The longer legs of the LEDs are the anodes or positive legs. These are the ones we connect to the micro:bit pins. The shorter legs are the cathodes, negative. They get connected to GND through a resistor.



#### **Programming - Digital**

We can turn the LEDs off by using the  $write\_digital()$  method and supplying either a 1 (HIGH) or 0 (LOW) in the brackets to specify what we want to do.

This program blinks each of the LEDs in turn.

```
from microbit import *
pin0.write_digital(0)
pin1.write_digital(0)
pin2.write_digital(0)
sleep(1000)
while True:
    pin0.write_digital(1)
    sleep(500)
    pin0.write_digital(0)
    pin1.write_digital(1)
    sleep(500)
    pin1.write_digital(1)
    sleep(500)
    pin1.write_digital(0)
    pin2.write_digital(1)
    sleep(500)
    pin2.write_digital(1)
    sleep(500)
    pin2.write_digital(0)
```

#### **Programming - Analog**

We use the **write\_analog()** with a value from 0 to 1023. This creates a PWM signal (pulse width modulation) on the pin that makes the LEDs fade in and out very slowly.

```
from microbit import
pin0.write digital(0)
pin1.write_digital(0)
pin2.write_digital(0)
sleep(1000)
while True:
    for i in range(0,1024):
        pin0.write_analog(i)
        pin1.write_analog(i)
        pin2.write_analog(i)
        sleep(10)
    for i in range(1023,-1,-1):
        pin0.write_analog(i)
        pin1.write_analog(i)
        pin2.write_analog(i)
        sleep(10)
```

#### **BBC Microbit**

**BBC** micro:bit

Arduino

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
- \* Introduction
- \* Buzzer With MicroPython
- ★ LEDs With MicroPython
- \* Connecting micro:bits Together
- 🛨 Extra Buttons
- X Knock Sensor
- \* Rotary Encoder
- \* Potentiometer
- \* Soft Potentiometer
- \* Flex Sensor
- \* Tilt Sensor
- 大 Reed Switch 大 More Buttons
- \* Temperature Sensor
- 大 7 Segment Display
- \* Reflectance Sensor
- ★ Driving A Motor
- \* Shift Register
- 大 Shifting In 大 Neopixels
- \* IR Break Beam Sensor
- ★ DIY MIDI Out
- \* PCF8574A Port Expander
- ★ 16x2 Character LCD Display
- \* SNES Controller
- + MicroPython Breakout Boards
- + MicroPython Exercises
- + MicroPython Pi Accessories
- + MicroPython Bit:Bot
- + MicroPython Bit:Commander
- + MicroPython Projects
- + MicroPython Visual Basic
- + Other Odds & Ends







11/17/2018 multiwingspan

### **Challenge**

Using 3 LEDs, you can count up to 7 in binary. The following table shows you how we can represent these using 3 bits (3 binary place values).

Denary	Binary
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

The following pseudocode will give you a one or zero in the variables fours, twos and units. Your leftmost LED is the fours LED, then twos, then units.

```
denary ← any integer from 0 to 7 included fours ← denary // 4 remainder ← denary % 4 twos ← remainder // 2 units ← remainder % 2
```

Pages designed and coded by MHA since 2003 | Valid HTML 4.01(Strict) | CSS