

# Christmas Micro:bit

## Objectives

To understand how run multiple processes simultaneously with your micro:bit. (Screen, sound and LEDs)

## Success Criteria

Demonstrate multiple simultaneous processes running on your Micro:bit

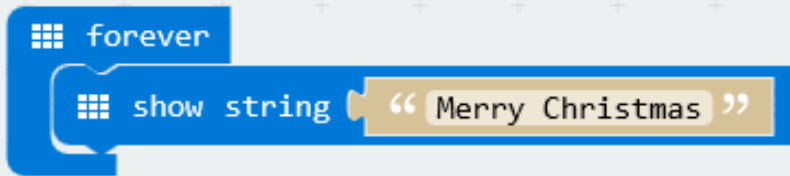
## Literacy Objective

Be able to enter commands into a computer with accuracy

# The screen

## Starter for 10

Create a Simple message to display on your Micro:bit screen



```
1 from microbit import *
2
3 while True:
4     display.scroll("Merry Christmas")
```

# Making Music

## Simple Stuff - Python

- You must import the library with Python ***Import music***
- There are some pre-made songs try ***music.play(music.NYAN)***
- See ***[HERE](#)*** for a song list

## Writing your own

- Each note has a name (like c#), an octave (how high or low) and a duration.
- Octaves are 0 for the lowest, 4 contains middle C and 8.
- Duration is expressed in numbers, the bigger the number the longer the duration. (4 will last twice as long as two for example).
- Micropython also supports the note R which means rest
- The definition and scope of an octave conforms to the table listed [on this page about scientific pitch notation](#).
- ***NOTE[octave][:duration]***

## Extra

- Tempo and beats - A number of ticks (expressed as an integer) constitute a beat.
- Each beat is to be played at a certain frequency per minute (expressed as the more familiar BPM - beats per minute - also as an integer).
- ***music.set\_tempo(ticks=4, bpm=120)***

# Making Music Python

## Python

For this example I have Pin 0 and ground connected to the speaker and we will use this piece of music

G, A, G, E  
G, A, G, E  
D, D, B..  
C, C, G..  
A, A, C, B, A, G.. A, G, E..  
A, A, C, B, A, G.. A, G, E..  
D, D, F, D, B, C, C, E..  
C, G, E, G..  
F, D, C..

## Notation

- Working on the basis of a duration of 8 being a full not I've come up with the following Python Code.
- I'm no musician!! So [HERE](#) is the score for you to improve I based it on the score in [THIS](#) video.

```
1 from microbit import *
2 import music
3 music.set_tempo(bpm=68)
4 #Silent Night in C
5
6 SilentNight = [
7 "G4:6","A4:2","G4:4","E4:10","G4:6","A4:2","G4:4",
8 "E4:10","D5","B4","C5",
9 "G4","A4:6","C5:6","B4:1","A4:2","G4:4","A4:2","G4:4",
10 "E4:10","A4:6","C5:6","B4:1","A4:4","G4:6","A4:2","G4:4",
11 "E4:10","D5:8","F5:6","D5:2","B4:6","C5:6",
12 "E5:10","C5:6","G4:2","E4:4","G4:6","F4:2","D4:4","C4:10"
13 ]
14 music.play(SilentNight)
15
```

# Making Music PXT

## PXT

- The same notation in PXT would look like below.
- Please note the PXT implementation of sound is not as clean as Python – you may need to fiddle with the notes to make it sound right on the speaker



```
set tempo to (bpm) 68
play tone G4 for 1/2 beat
play tone G4 for 1/4 beat
play tone A for 1/4 beat
play tone G4 for 1/2 beat
play tone E4 for 1 beat
play tone E4 for 1/4 beat
play tone G4 for 1/2 beat
play tone G4 for 1/4 beat
play tone A4 for 1/4 beat
play tone G4 for 1/2 beat
play tone E4 for 1 beat
play tone E4 for 1/4 beat
```

# The Lights

- For this example I am using a 2 programmable LEDs in parallel. This could easily be a strip of Neo-pixels and would work in the same way.
- The only limit is do not use more than 8 LEDs without a separate power supply

```
from microbit import *
import neopixel
from random import randint
np = neopixel.NeoPixel(pin1, 2)
while True:

    for pixel_id in range(0, len(np)):
        red = randint(0, 255)
        green = randint(0, 255)
        blue = randint(0, 255)

        # Assign the current LED a random red
        np[pixel_id] = (red, green, blue)

    # Display the current pixel data on the
    np.show()
    sleep(50)
```

The image shows a block-based code editor with the following sequence of blocks:

- on start** block
- set NP to** NeoPixel at pin P1 with 2 leds as RGB (GRB format)
- NP clear** block
- item show** block
- while true** loop:
  - do** block containing:
    - set RED to** pick random 0 to 255
    - set GREEN to** pick random 0 to 255
    - set BLUE to** pick random 0 to 255
  - item set pixel color at 0 to** (red RED green GREEN blue BLUE)
  - NP show** block
  - pause (ms)** 1000

# All together now!

## MicroPython

- MicroPython has a feature that is not well documented that allows you to run processes in the background
- **wait=False** (this can be set to True or False – if False the Micro:bit won't wait for the process to finish before running the next line of code)
- **Loop = True** will force a process to be repeated even if it's not inside a for or while loop

## PXT

The same result can be completed using the “Run in Background” block in PXT

```
32
33 music.play(jingle,wait=False,loop=True)
34 display.scroll("merry christmas",wait=False,loop=True)
35
36 np = neopixel.NeoPixel(pin1, 8)
37 while True:
38
39     for pixel_id in range(0, len(np)):
40         red = randint(0, 255)
41         green = randint(0, 255)
42         blue = randint(0, 255)
43
44         # Assign the current LED a random red, green and
```

The screenshot shows a PXT code editor with two main blocks. The first is a 'run in background' block containing a 'while true' loop with a 'show string' block (displaying 'Merry Christmas') and a 'show' block. The second is an 'on start' block containing a 'set NP' block (NeoPixel at pin P1 with 2 leds as RGB (GRB format)), a 'clear' block, a 'show' block, a 'while true' loop with 'set RED' (pick random 0 to 255), 'set GREEN' (pick random 0 to 255), 'set BLUE' (pick random 0 to 255), 'set pixel color at 0 to red GREEN blue' blocks, a 'show' block, and a 'pause (ms) 1000' block. A third 'run in background' block is also visible, containing a 'set tempo to (bpm) 75' block and a 'while true' loop with a 'do' loop of 'play tone' blocks (Middle E, Middle E, Middle E, Middle E, Middle E, Middle E, Middle G, Middle C, Middle D, Middle E).

# Give it a go

## Task

Your turn - Try this

- Create a display on your screen that runs as a background processing and loops forever
- Write some music to play in the background on a repeat. Try and convert a simple nursery rhyme or carol. I have done Jingle Bells [HERE](#)
- Using a Neopixel or Programmable LED rotate some pretty colours.
- If you only have standard LEDs then you could use red and green and flash them alternatively.
- Make all this happen together

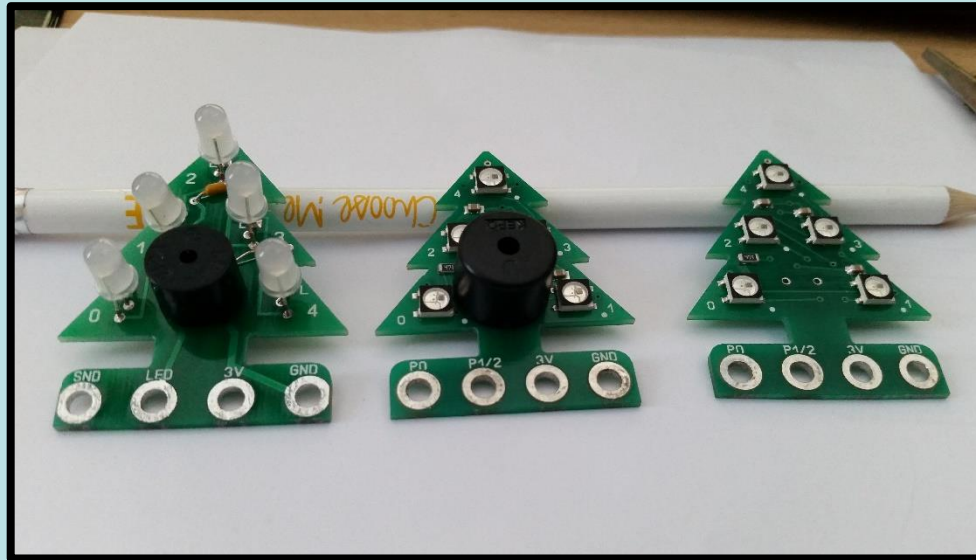
**Most of all – Be creative and have fun!!**



# Oh Christmas Tree

## PocketmoneyTronics Christmas Tree

- [www.microbitsandbobs.co.uk](http://www.microbitsandbobs.co.uk) have put this resource together in order to support buyers of the awesome <http://www.pocketmoneytronics.co.uk/> Christmas Tree.
- This is currently on [Kickstarter](#) (25/10/16) which I hope is very successful and I'm sure you will be able to buy them direct from Andrew Gale via his website really soon.
- At the time of writing, I am aware of 3 Micro:bit versions of the Christmas Tree.
- Solder yourself, Neopixel with buzzer and Neopixel without buzzer



**All the code here runs with any BBC Micro:bit with a piezo speaker and either standard programmable LED or neopixel ws2812 packages.**