## Lesson 27 – Activity Sheet

Getting Started

In a previous lesson we looked at how to control each pixel, turn them on and adjust the colour. In the activity you will combine the Halo and a **temperature** reading to create a real time temperature display.

A picture containing light, sitting, table, indoor

Description automatically generated

The pixels in red indicate the amount of warmth or heat and the rest of the pixels are left blue.

The hotter the temperature the more pixels are coloured red. The colder the temperature the more pixels are lit blue.

The pixels represent a ratio between the hot and cold.

## **Part One**

Start a new program and add the following lines of code.

from microbit import \*

import neopixel

np = neopixel.NeoPixel(pin0, 24)

The third line of code enables the micro:bit to know that there are 24 pixels and they are accessible on Pin 0. This information is stored in a variable named *np*.

## **Part Two**

Next add the next three lines of the program code. Note the indentation levels used. This must be observed to ensure that the program runs correctly.

while True:

temp = temperature()-8

for pixel\_id in range(0, len(np)):

The first line creates a while loop which keeps the program running the next lines of code. This ensures that the program keeps checking the temperature and responding. This is what enables the program to be a real time temperature display, it adjusts the display in real time. The next line takes the temperature reading and stores it in a variable called *temp*. Note that a value of 8 is subtracted from the value to ensure that it is accurate. The last line checks how many pixels there are using the range function and then uses iteration to apply the next lines of code to each pixel.

## **Part Three**

Next we add the code to change each pixel in the interaction. This first line is applying the colour and turning each pixel red, this represents the temperature. The next line is used to turn the pixels on. Then the program uses another for loop to calculate the pixels that need to be turned blue. The current temperature is subtracted from the value stored in *np*, the total number of pixels, 24 in this program. For example, if the temperature is 18, then the calculation is 24 – 18 = 6. This means that 6 pixels are coloured blue. The pixels are coloured blue on line 4 and then turned on.

np[pixel\_id] = (255, 0, 0)

np.show()

for pixel\_id in range(0, len(np)-temp):

np[pixel\_id] = (0, 0, 255)

np.show()

## **Part Four**

The final section of the program code adds a short pause, in this program 5 seconds, and then clear the pixels. The code basically resets them. The program is looping so the process started again, a temperature reading is taken and then the pixels are coloured and turned on to represent the temperatures.

sleep(5000)

np.clear()

## **The Final Program**

The program below is a starter template for a **real time** temperature indicator:

from microbit import \*

import neopixel

from random import randint

np = neopixel.NeoPixel(pin0, 24)

while True:

temp = temperature()-8

for pixel\_id in range(0, len(np)):

np[pixel\_id] = (255, 0, 0)

np.show()

for pixel\_id in range(0, len(np)-temp):

np[pixel\_id] = (0, 0, 255)

np.show()

sleep(5000)

np.clear()

## Success Criteria

* Create the program to build a real time temperature sensor
* Adjust the temperature reading to test the hardware

## Pro-tip

Ensure that the indentations are correct

## Test Time

Add the battery pack to the Halo and then test the program by adjusting the temperature. If it is a warm day, take the Halo outside, if you need to cool the Halo down then place it in a fridge for a few minutes.

## Stretch Tasks

* Adjust the timing on the last line
* Customise the colour of the pixels
* Edit the line for pixel\_id in range(0, len(np)-temp) to for pixel\_id in range(0, len(np)-temp/2). This will increase the maximum temperature from 24 to 48, which means the display can represent hotter temperatures.
* Can you adjust the previous line if code to reflect lower temperatures, such as negative temperatures?

## Final Thoughts

The building of a real time temperature sensor uses lots of the skills and elements from the previous lessons: functions, selection, displaying text and displaying images.