# Lesson 23 – Activity Sheet

# Getting Started

In this activity you will use the micro:bit’s built-in **temperature** sensor to retrieve a temperature reading in **Celsius** and then display the value across LED matrix. The learners then adjust the final value to make it more accurate.

## **Taking a Temperature Reading**

This program uses a function, microbit.temperature() to take the temperature reading of the micro:bit. This is usually influenced by the CPU, which produces a lot of heat and will affect the temperature. Enter the program code below and download it. Shake the micro:bit to display the current temperature.

from microbit import \*

while True:

if accelerometer.was\_gesture('shake'):

display.scroll(temperature())

Alter the program so that the temperature is displayed when Button A is pressed.

**Adjusting the Accuracy**

To make the reading more accurate, use a thermometer to take a reading of the temperature of the room. Compare this with the value that the micro:bit is displaying. You can then subtract the difference to return a more accurate reading.

For example, if the micro:bit displays 24 degrees and the room is 18 degrees, then the program needs to subtract 6 degrees. Edit the line of code and add the value to subtract.

display.scroll(temperature()-6)

Make the changes to the program and then download and run

**A Real Time Temperature Display**

The program can now be adjusted to respond to the temperature in **real time**. This means that you do not have to shake the micro:bit each time to display the temperature. Instead, the program takes a temperature reading and then adjusts and updates the display accordingly.

See if you can program it:

while True:

temp = temperature()

if temp xx

xxxx

elif temp xx

xxxx

else:

xx

## **The Complete Program Code**

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

from microbit import \*

while True:

temp = temperature()

if temp > 26:

display.scroll("Hot")

elif temp > 6 < 15:

display.scroll("Just right")

else:

display.scroll("Too cold")

* Remember to adjust the temperature values to reflect your environment
* The line elif temp > 6 < 15: is used to check if the temperature is greater than 6 degrees but less than 15
* You can add additional elif statements to measure extra temperature readings

## Success Criteria

* Program the temperature program to respond to a ‘shake’
* Adjust the accuracy of the temperature reading
* Program a basic real time temperature display, that responds to two readings
* Program a real time temperature display, that responds to three or more readings
* Add images or text to respond to readings

## Pro-tip

Remember that the temperature sensor is located on the micro:bit and picks up the heat from CPU. In order to get an accurate reading ensure that you adjust the final value by subtracting the difference between the reading and the room temperature.

## Test Time

When running the program adjust the temperature values to reflect your environment. If it is summer then the top temperatures will be higher, if it is winter then they will be lower. Download the program to the micro:bit and then use the battery pack to make the project portable.

## Stretch Tasks

* Display the temperature on the LEDs so that you can test that the program is working correctly
* Add the battery pack and take the micro:bit to a colder location, does the display change? Place it in the fridge!
* Add delays between each display.scoll() so that it is easier to view the temperature
* Instead of using text add images to reflect the temperatures. Try display.show(Image.HAPPY) instead
* Wire up the sound / music and create an alarm that is triggered when a particular temperature is reached

## 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, displaying images.