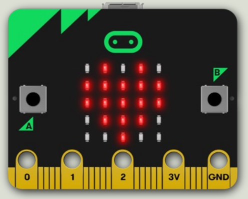
https://www.halvorsen.blog



micro:bit and TMP36 Temperature Sensor

Hans-Petter Halvorsen

Contents

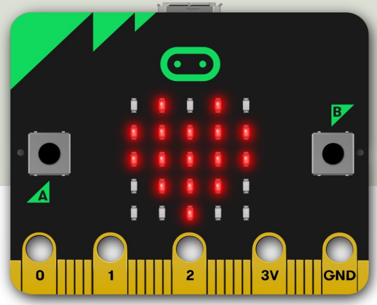
- Introduction to micro:bit and Python/MicroPython
- Using the built-in Temperature Sensor
- micro:bit I/O Pins
  - Analog and Digital Pins used for communication with external components, like LEDs, Temperature Sensors, etc.
- Using an external TMP36 Temperature Sensor
  - Python Examples

Introduction to micro:bit

Hans-Petter Halvorsen

Table of Contents

micro:bit



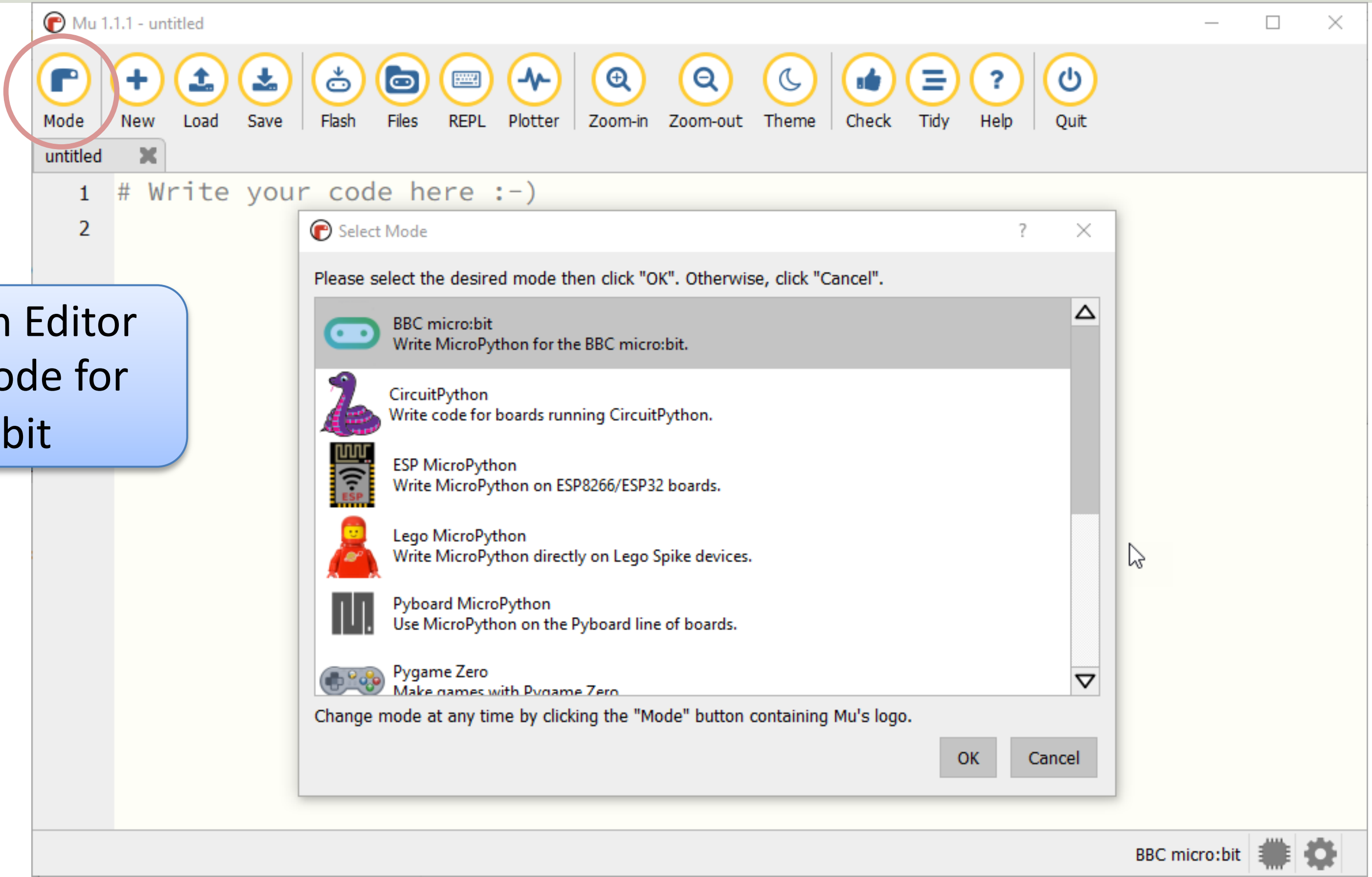
- micro:bit is a small microcontroller
- micro:bit is smaller than a credit card
- Price is about 150-400NOK ($15-30)
- It can be used by kids and students to learn programming and technology
- micro:bit can run a special version of Python called MicroPython
- MicroPython is a down-scaled version of Python

https://microbit.org

Mu Python Editor

- Mu is a Python code editor for beginners
- It is tailor-made for micro:bit programming
- Mu has a “micro:bit mode” that makes it easy to work with micro:bit, download code to the micro:bit hardware, etc.
- Mu and micro:bit Tutorials: https://codewith.mu/en/tutorials/1.0/microbit

Mu Python Editor



```
1 # Write your code here :-)
2
```

Please select the desired mode then click "OK". Otherwise, click "Cancel".

BBC micro:bit
Write MicroPython for the BBC micro:bit.

CircuitPython
Write code for boards running CircuitPython.

ESP MicroPython
Write MicroPython on ESP8266/ESP32 boards.

Lego MicroPython
Write MicroPython directly on Lego Spike devices.

Pyboard MicroPython
Use MicroPython on the Pyboard line of boards.

Pygame Zero
Make games with Pygame Zero

Change mode at any time by clicking the "Mode" button containing Mu's logo.

The Mu Python Editor has built-in Mode for the micro:bit

Built-in Temperature Sensor

Hans-Petter Halvorsen

Table of Contents

Temperature Sensor

- Micro:bit has a built-in Temperature Sensor (that is located on the CPU)
- This sensor can give an approximation of the air temperature.
- Just use the built-in `temperature()` function in order to get the temperature value from the sensor

Temperature Sensor

In order to read the temperature, you just use the built-in `temperature()` function:

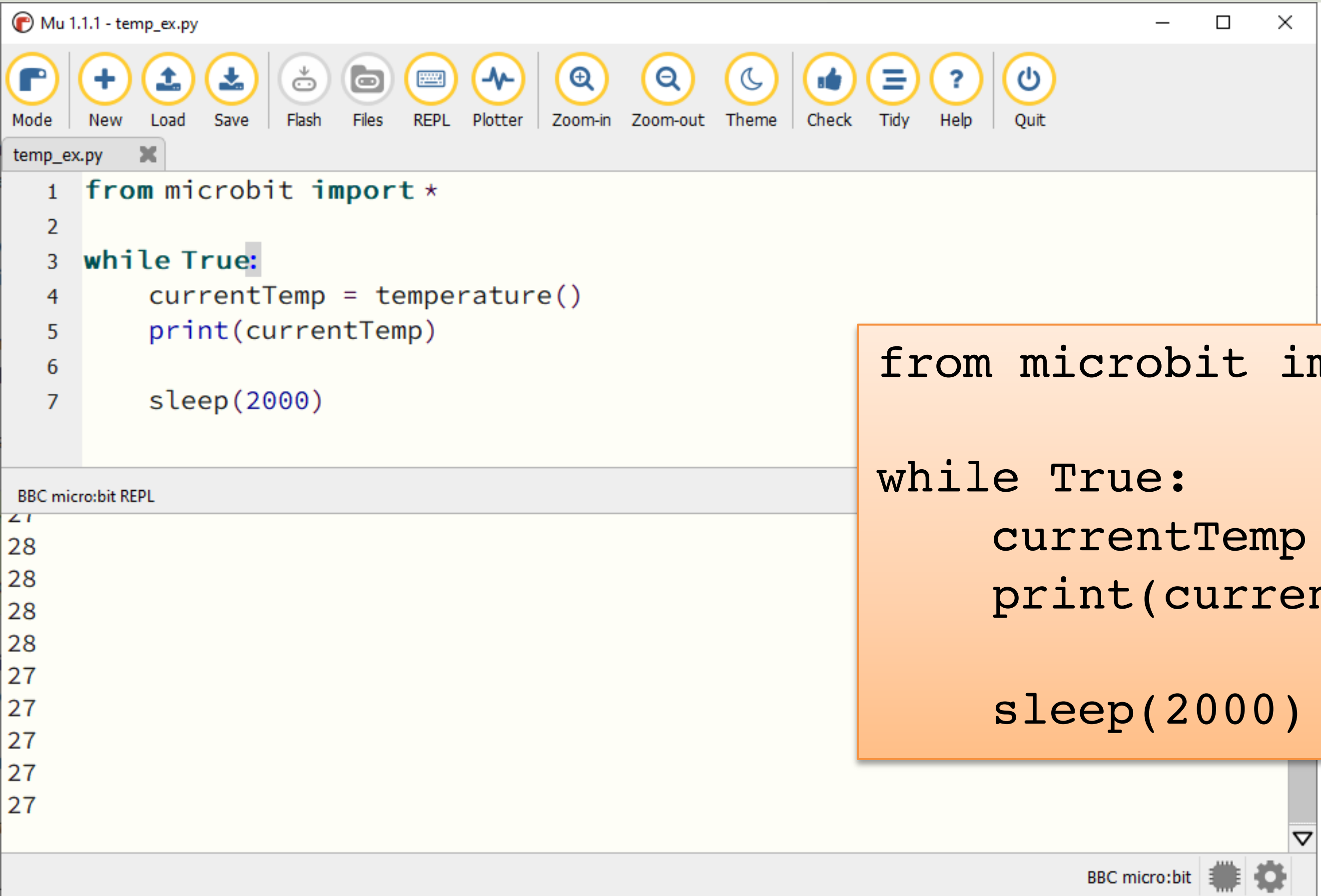
```
from microbit import *

currentTemp = temperature()
```

This examples displays the temperature on the LED matrix:

```
from microbit import *

while True:
    if button_a.was_pressed():
        display.scroll(temperature())
```

https://microbit.org/get-started/user-guide/features-in-depth/#temperature-sensor

Temperature Sensor



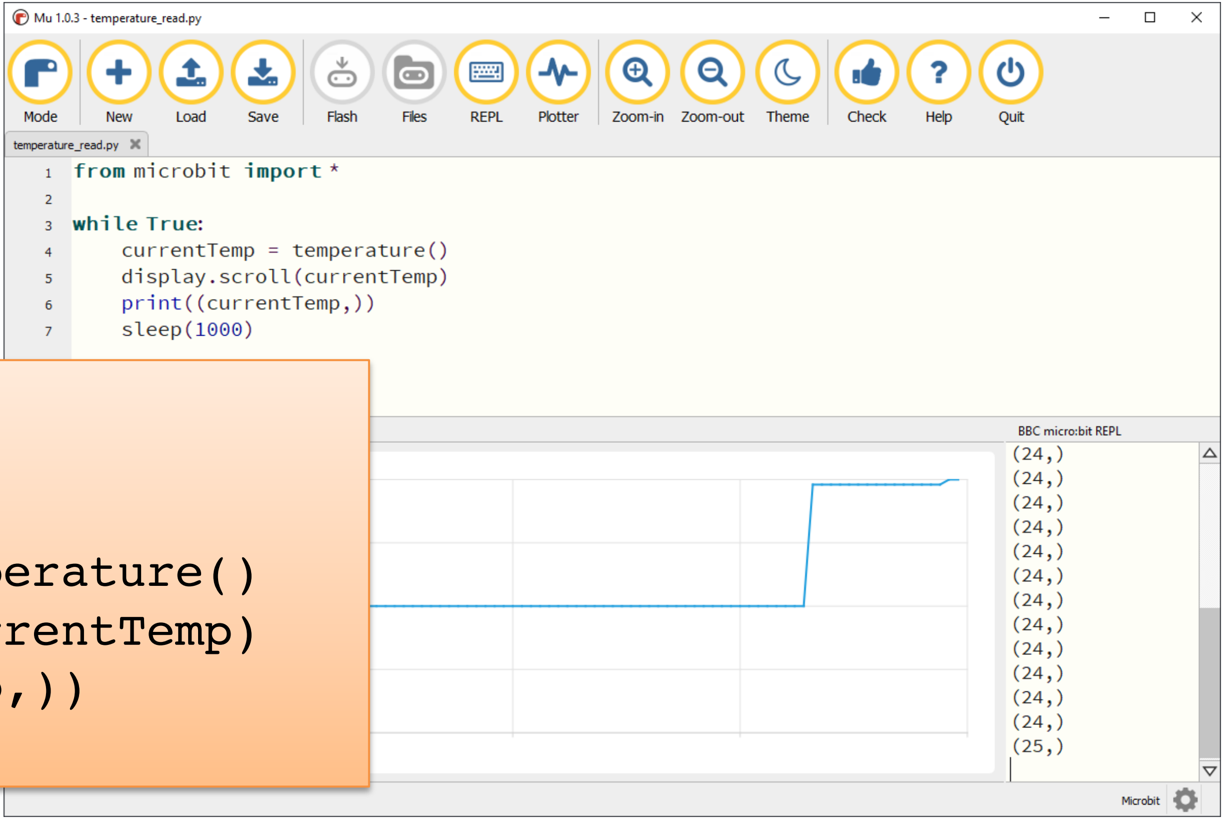
```
from microbit import *

while True:
    currentTemp = temperature()
    print(currentTemp)

    sleep(2000)
```

```
28
28
28
28
27
27
27
27
27
```

```
from microbit import *

while True:
    currentTemp = temperature()
    print(currentTemp)

    sleep(2000)
```

Temperature Sensor

```
from microbit import *

while True:
    currentTemp = temperature()
    display.scroll(currentTemp)
    print((currentTemp,))
    sleep(1000)
```



```
from microbit import *

while True:
    currentTemp = temperature()
    display.scroll(currentTemp)
    print((currentTemp,))
    sleep(1000)
```

```
(24,)
(24,)
(24,)
(24,)
(24,)
(24,)
(24,)
(24,)
(24,)
(24,)
(24,)
(24,)
(25,)
```

Display Min/Max Temperature

```
from microbit import *

currentTemp = temperature()
maxTemp = currentTemp
minTemp = currentTemp

while True:
    currentTemp = temperature()

    if currentTemp < minTemp:
        minTemp = currentTemp
    if currentTemp > maxTemp:
        maxTemp = currentTemp

    if button_a.was_pressed():
        display.scroll(minTemp)
    elif button_b.was_pressed():
        display.scroll(maxTemp)
    else:
        display.scroll(currentTemp)

    print((currentTemp, minTemp, maxTemp))
    sleep(2000)
```

If you do nothing, the LED matrix shows the Current Temperature.

If you click A Button, the Minimum Temperature for the period (since you started the program/turned on the Micro:bit) is shown on the LED matrix

If you click B Button, the Maximum Temperature for the period (since you started the program/turned on the Micro:bit) is shown on the LED matrix

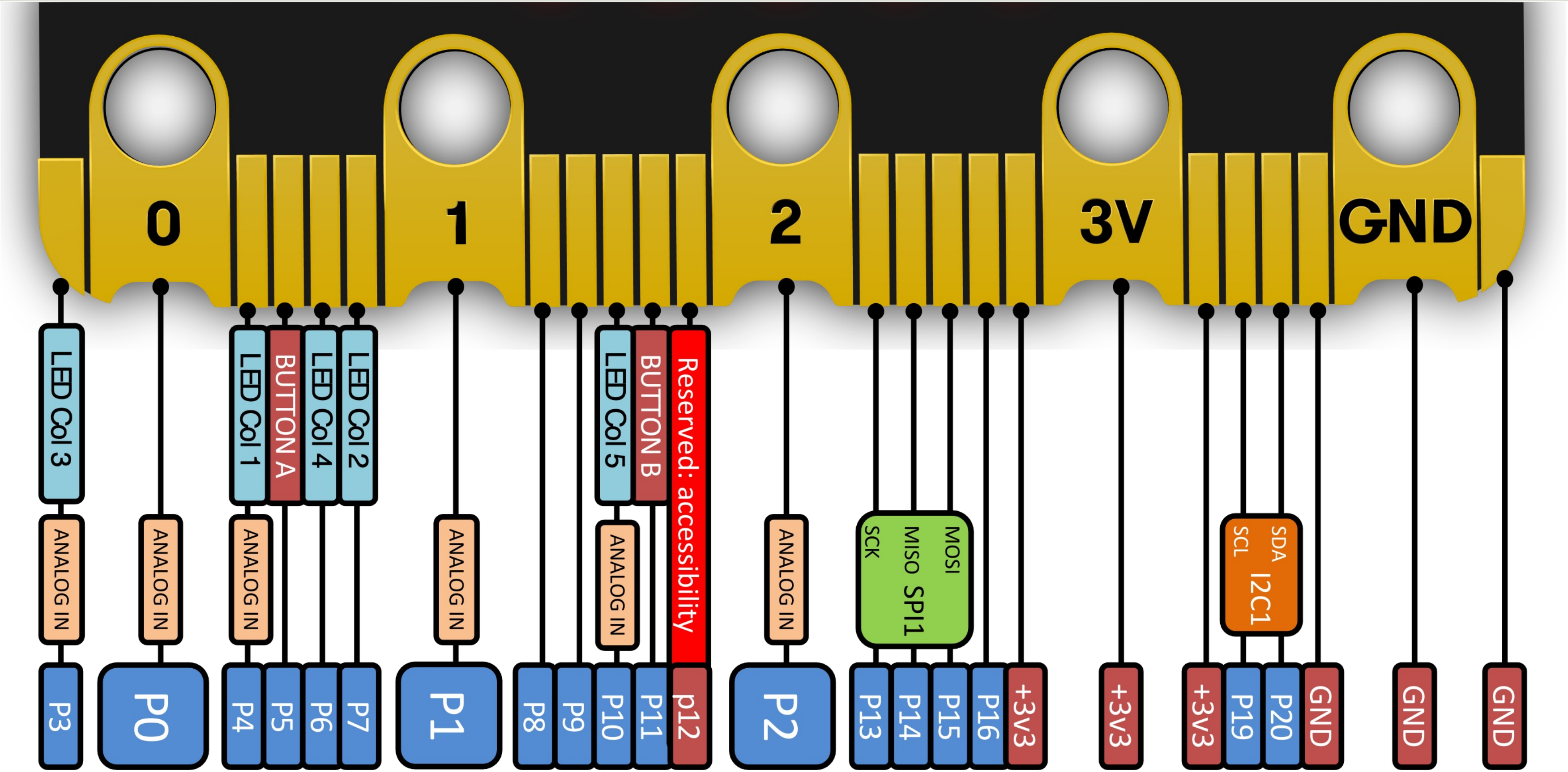
micro:bit I/O Pins

Hans-Petter Halvorsen

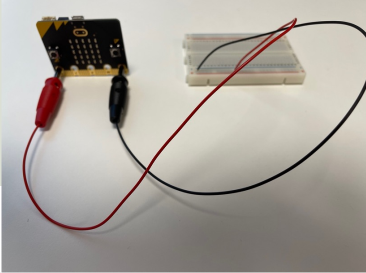
Table of Contents

https://microbit.pinout.xyz/

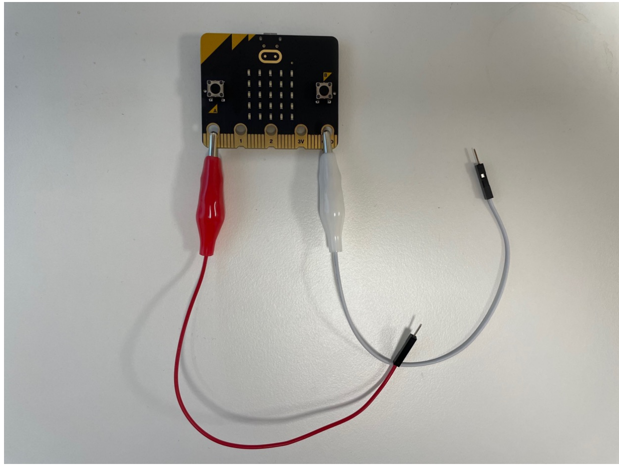
micro:bit I/O Pin Overview



I/O Pins



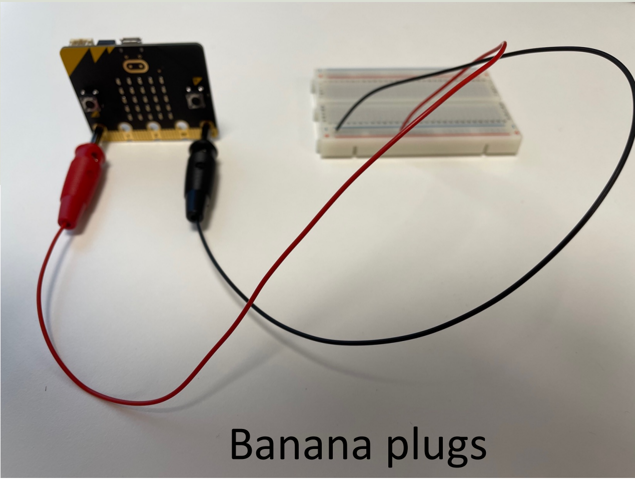
- We use the I/O pins to connect external components like LEDs, different types of Sensors, etc.
- You can use 4mm Banana plugs or Alligator/Crocodile clips
- Typically, you also want to use a Breadboard

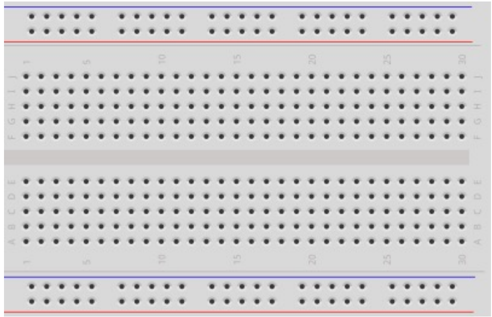


https://tech.microbit.org/hardware/edgeconnector/

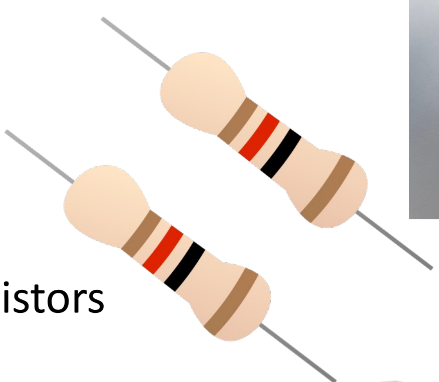
https://makecode.microbit.org/device/crocodile-clips

Component Examples

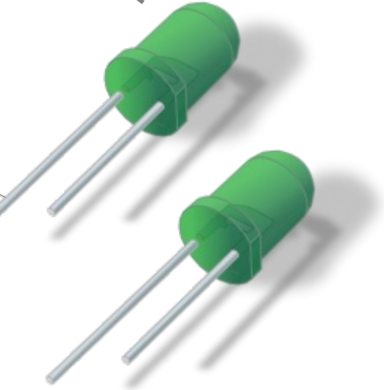

Banana plugs


Breadboard

Temperature Sensor

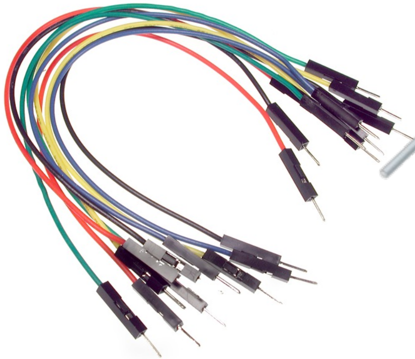

Resistors

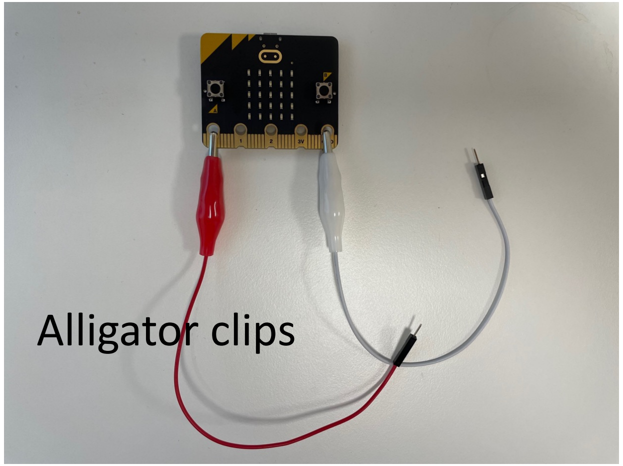
LEDs



Multimeter

Wires




Alligator clips

Types of I/O Pins

- Analog/Digital Input/Output Pins
- Pulse Width Modulation (PWM)
- SPI
- I2C
- UART (used for serial communication)

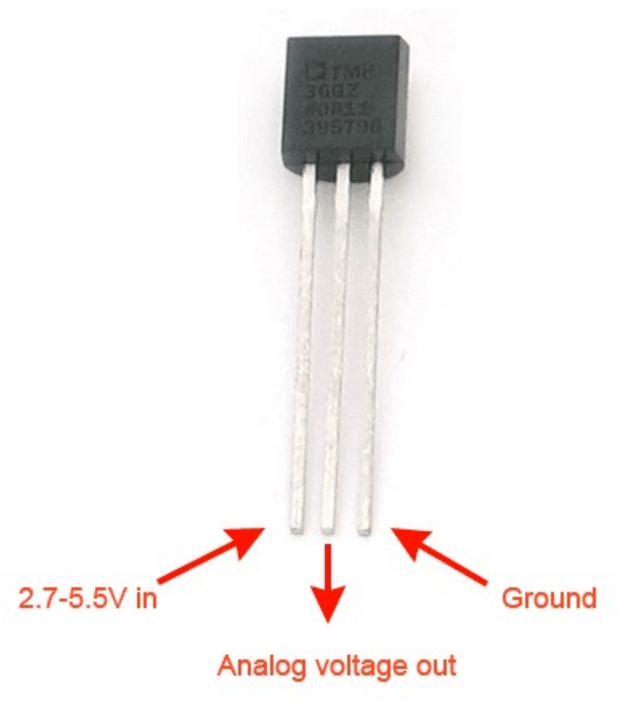
We will only use an Analog Input pin in this Tutorial

TMP36 Temperature Sensor

Hans-Petter Halvorsen

Table of Contents

TMP36 Temperature Sensor

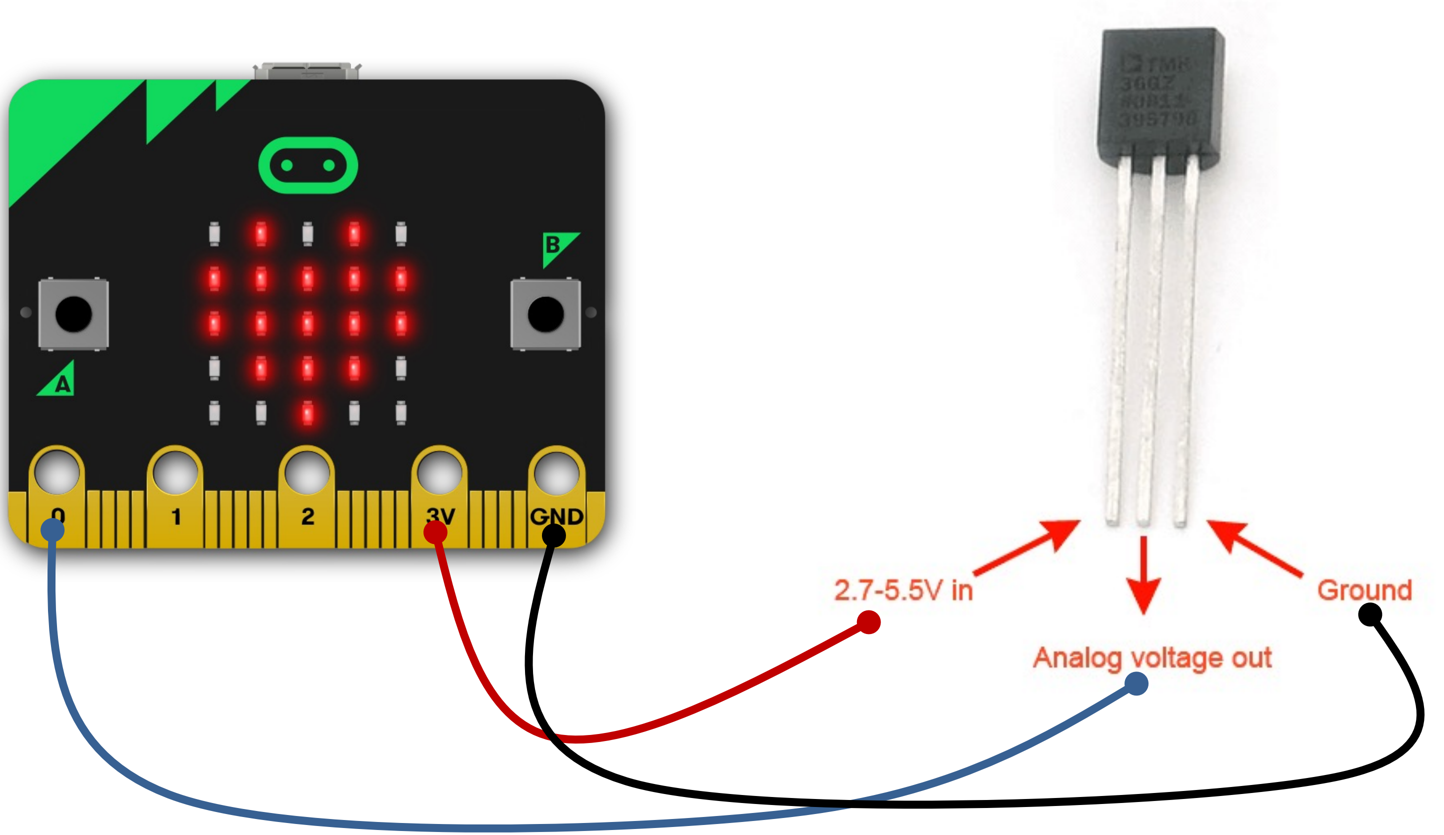


A Temperature sensor like TM36 use a solid-state technique to determine the temperature.

They use the fact as temperature increases, the voltage across a diode increases at a known rate.

https://learn.adafruit.com/tmp36-temperature-sensor

Wiring

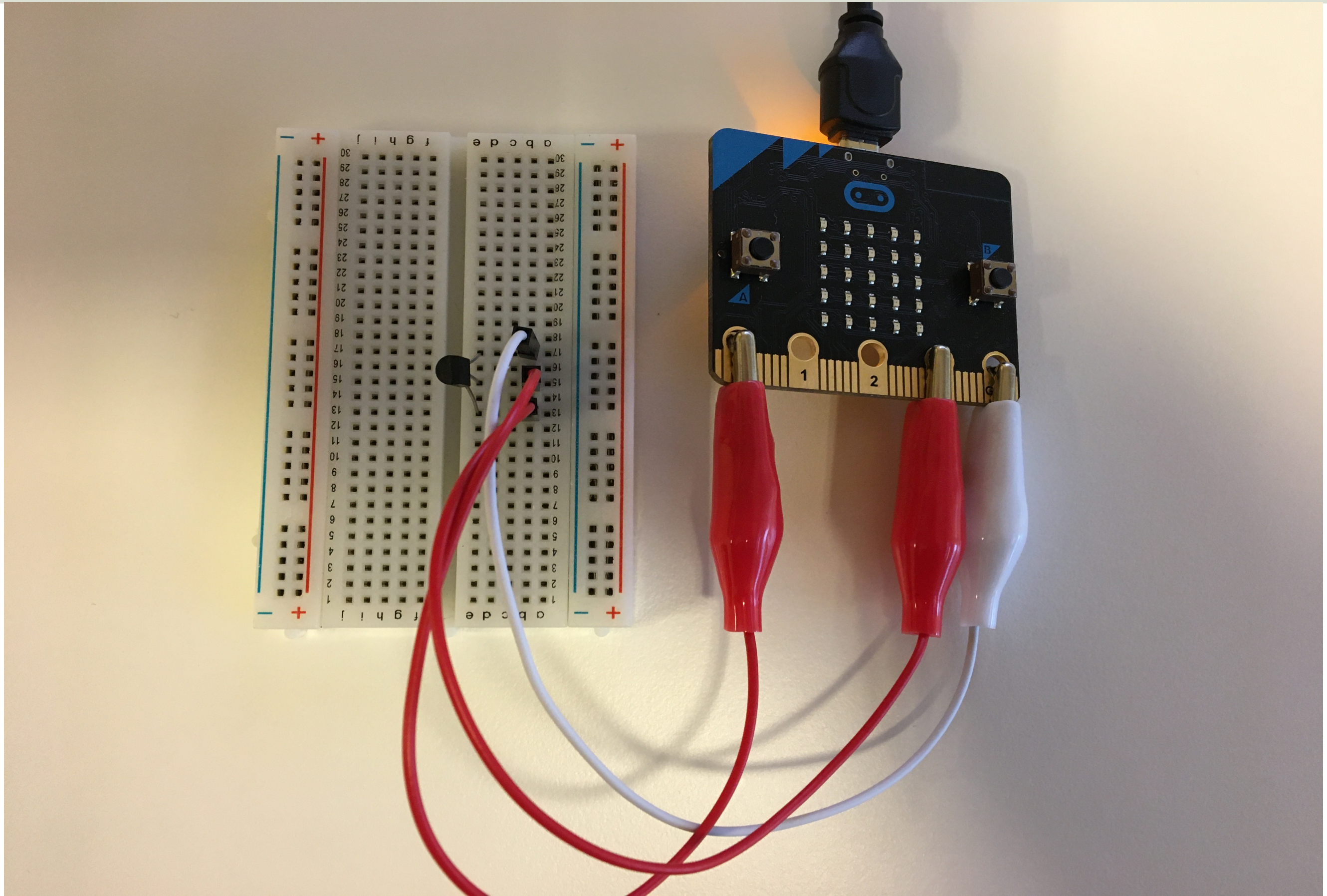


2.7-5.5V in

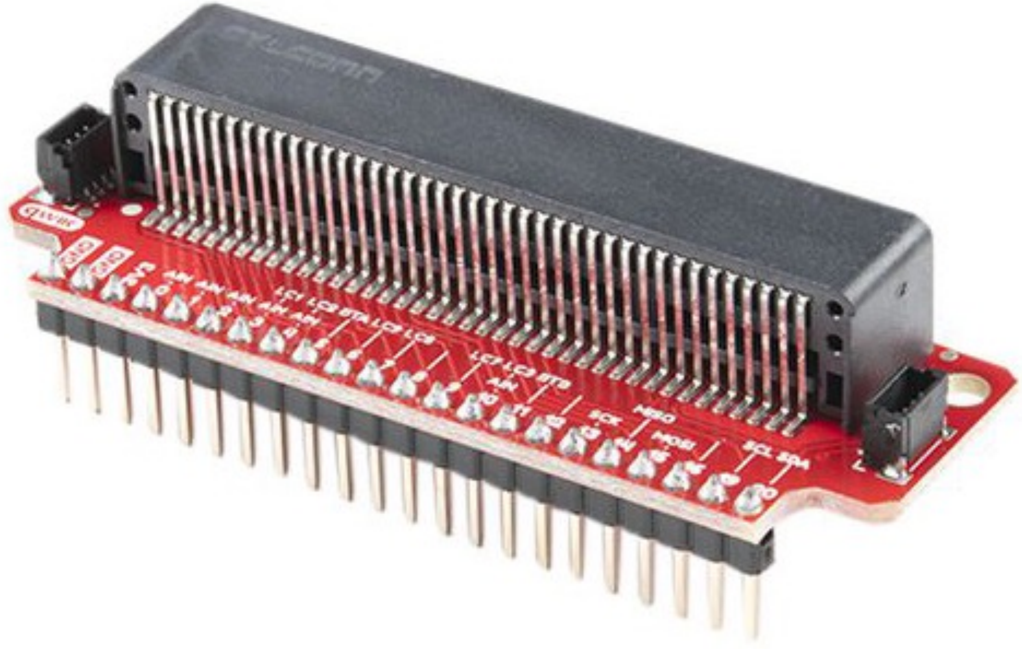
Analog voltage out

Ground

Breadboard and Crocodile clips



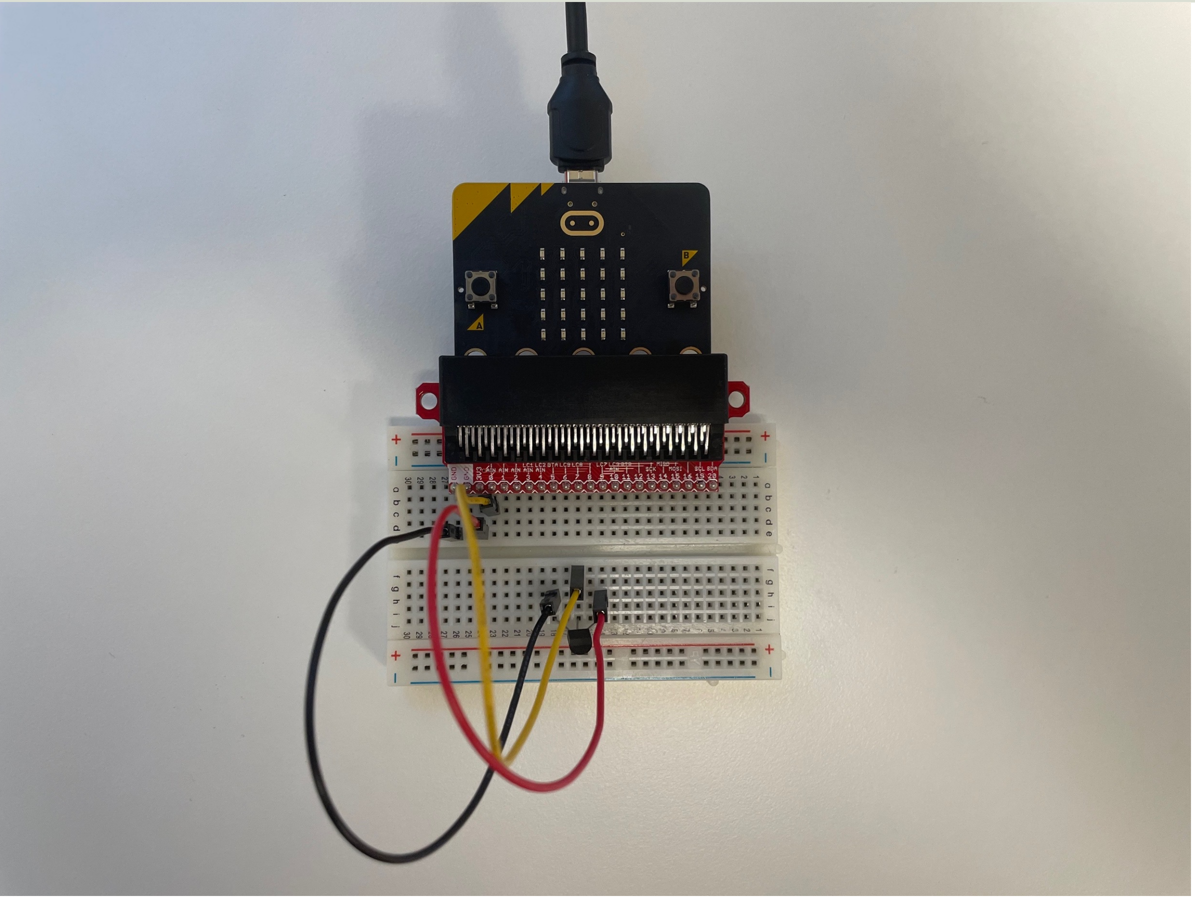
Adapter Breakout Board for micro:bit



We can also use an **Adapter Breakout Board for micro:bit** instead of Alligator/Crocodile clips

This makes it easier to wire for more advanced circuits and use of more in inputs/outputs pins

Adapter Breakout Board for micro:bit



Here you see see the wirings using an Adapter Breakout Board for micro:bit

Python

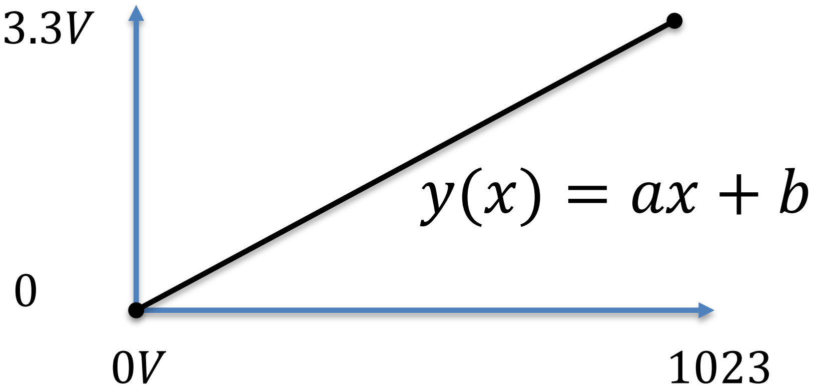
```
from microbit import *

while True:
    adc = pin0.read_analog()
    display.scroll(adc)
    sleep(5000)
```

ADC Value to Voltage Value

Analog Pins: The the built-in analog-to-digital converter (ADC) on micro:bit is 10bit, producing values from 0 to 1023.

The function `pin0.read_analog()` gives a value between 0 and 1023. It must be converted to a Voltage Signal 0 - 3.3v

ADC = 0 -> 0v
ADC = 1023 -> 3.3v



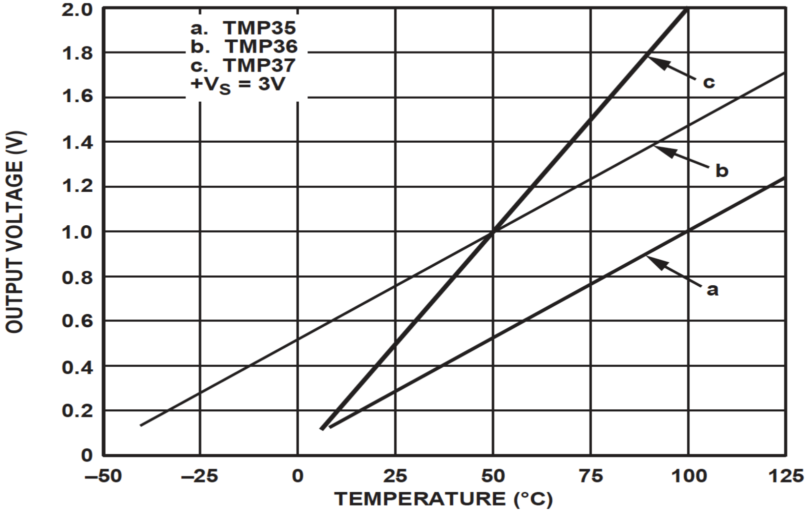
This gives the following conversion formula:

$$y(x) = \frac{3.3}{1023}x$$

Python

```
from microbit import *

while True:
    adc = pin0.read_analog()
    volt = (3.3/1023)*adc
    display.scroll(volt)
    sleep(5000)
```

Voltage to degrees Celsius



Convert form Voltage (V) to degrees Celsius

From the Datasheet we have:

$$(x_1, y_1) = (0.75V, 25°C)$$
$$(x_2, y_2) = (1V, 50°C)$$

There is a linear relationship between Voltage and degrees Celsius:

$$y = ax + b$$

We can find a and b using the following known formula:

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$$

This gives:

$$y - 25 = \frac{50 - 25}{1 - 0.75}(x - 0.75)$$

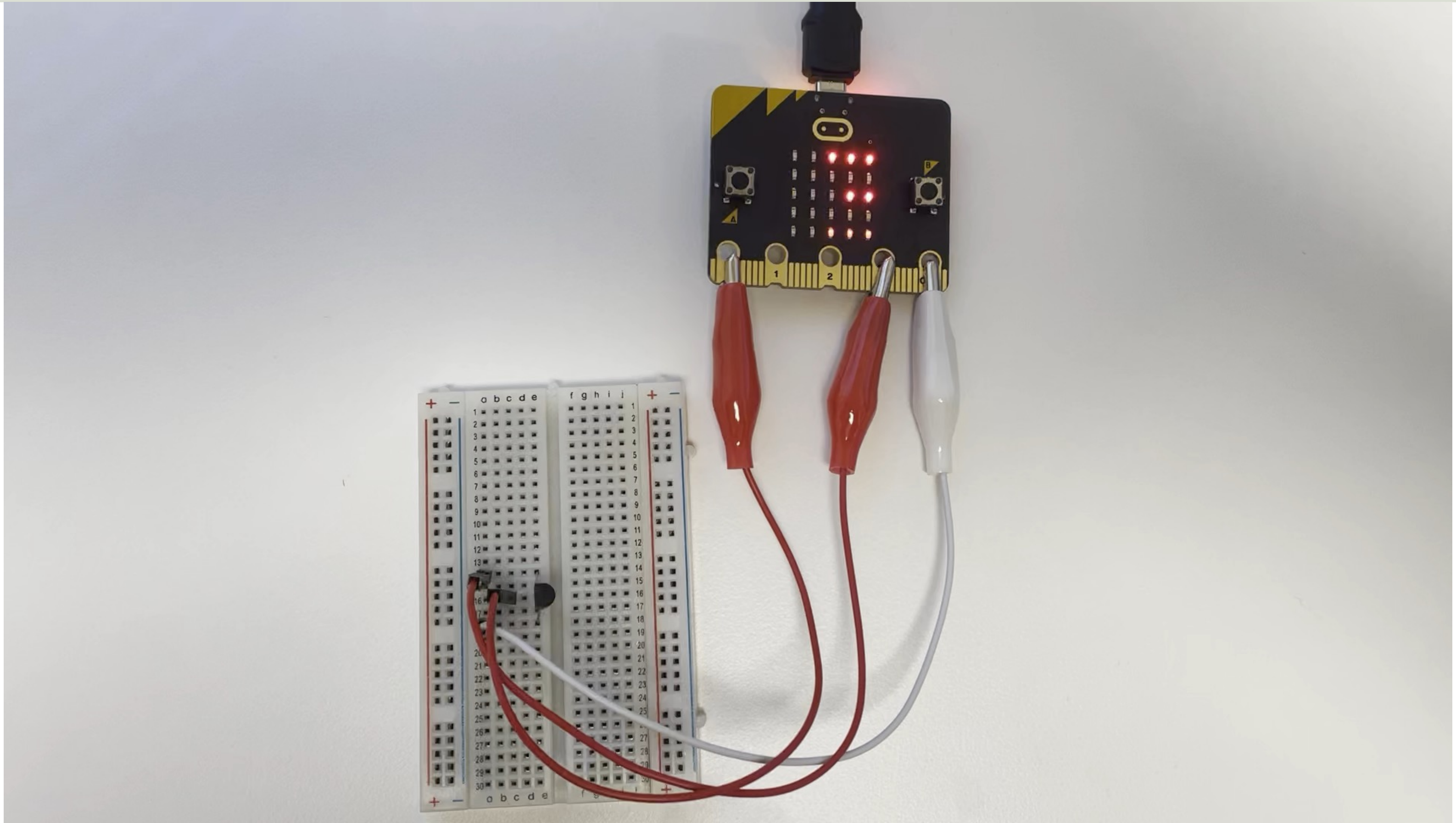
Then we get the following formula:

$$y = 100x - 50$$

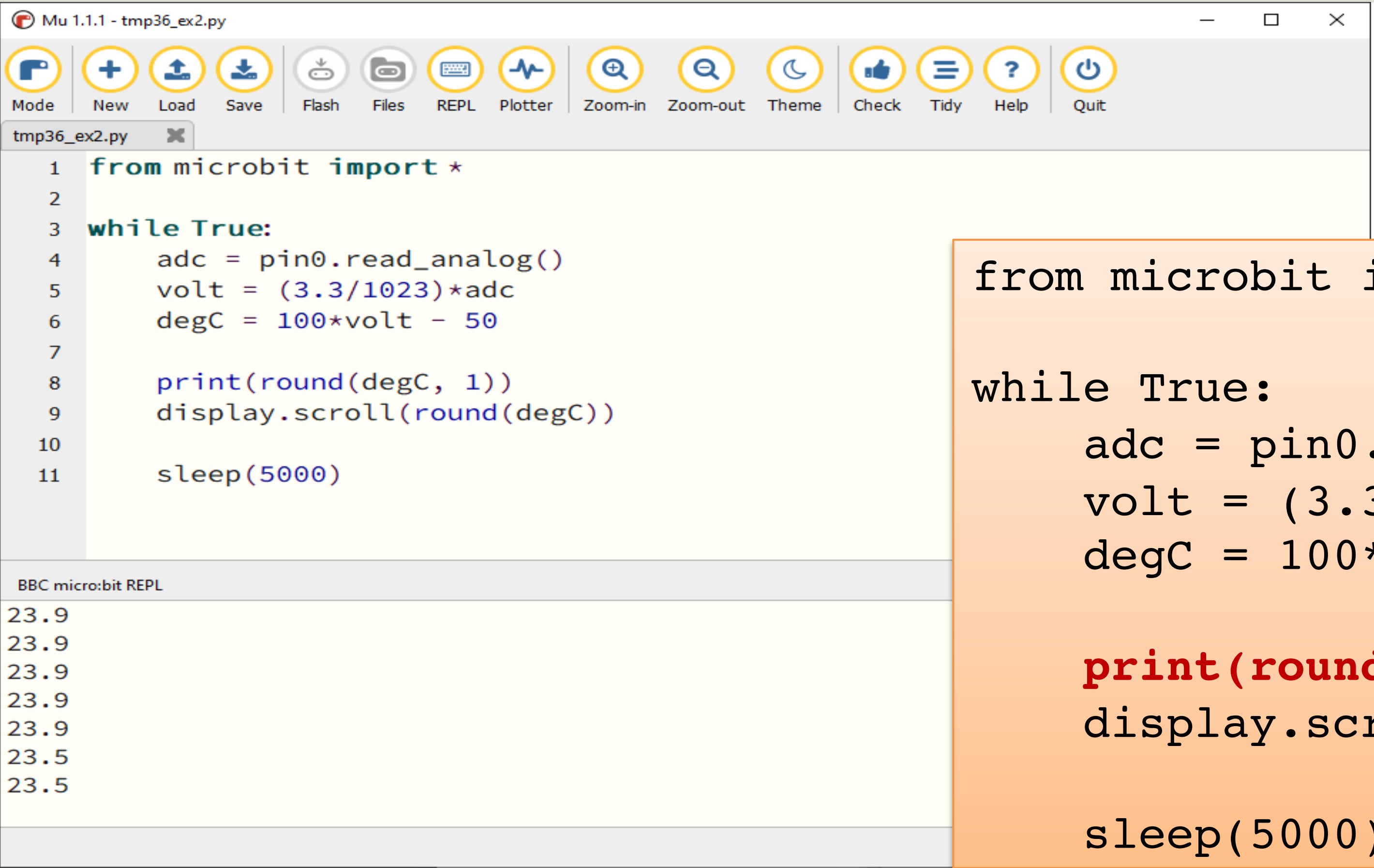
Python

```
from microbit import *

while True:
    adc = pin0.read_analog()
    volt = (3.3/1023)*adc
    degC = 100*volt – 50
    display.scroll(round(degC))
    sleep(5000)
```

Results



Printing to REPL

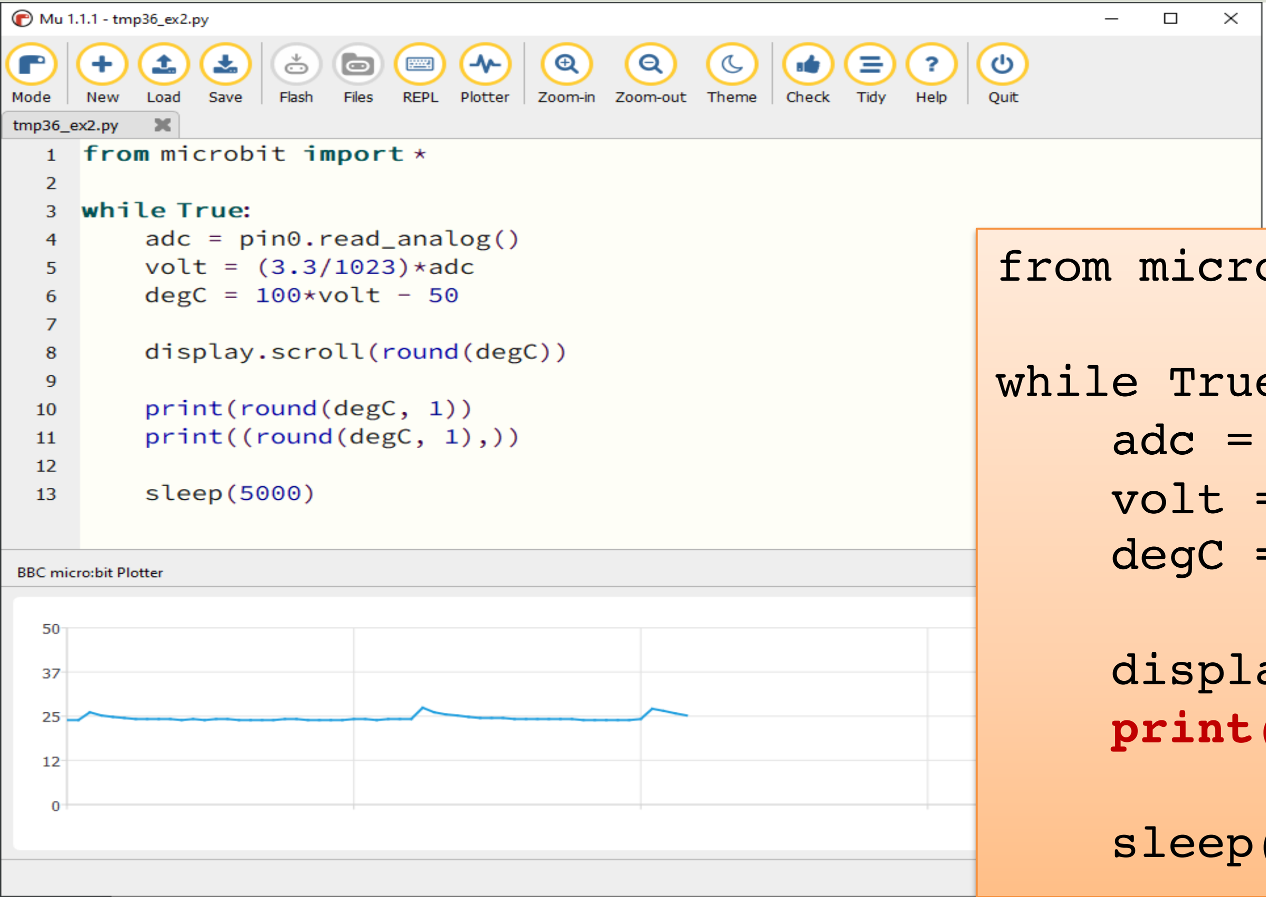


```
from microbit import *

while True:
    adc = pin0.read_analog()
    volt = (3.3/1023)*adc
    degC = 100*volt - 50

    print(round(degC, 1))
    display.scroll(round(degC))

    sleep(5000)
```

```
23.9
23.9
23.9
23.9
23.9
23.5
23.5
```

```
from microbit import *

while True:
    adc = pin0.read_analog()
    volt = (3.3/1023)*adc
    degC = 100*volt - 50

    print(round(degC, 1))
    display.scroll(round(degC))

    sleep(5000)
```

Plotting



```python
from microbit import *

while True:
    adc = pin0.read_analog()
    volt = (3.3/1023)*adc
    degC = 100*volt - 50

    display.scroll(round(degC))

    print(round(degC, 1))
    print((round(degC, 1),))

    sleep(5000)
```

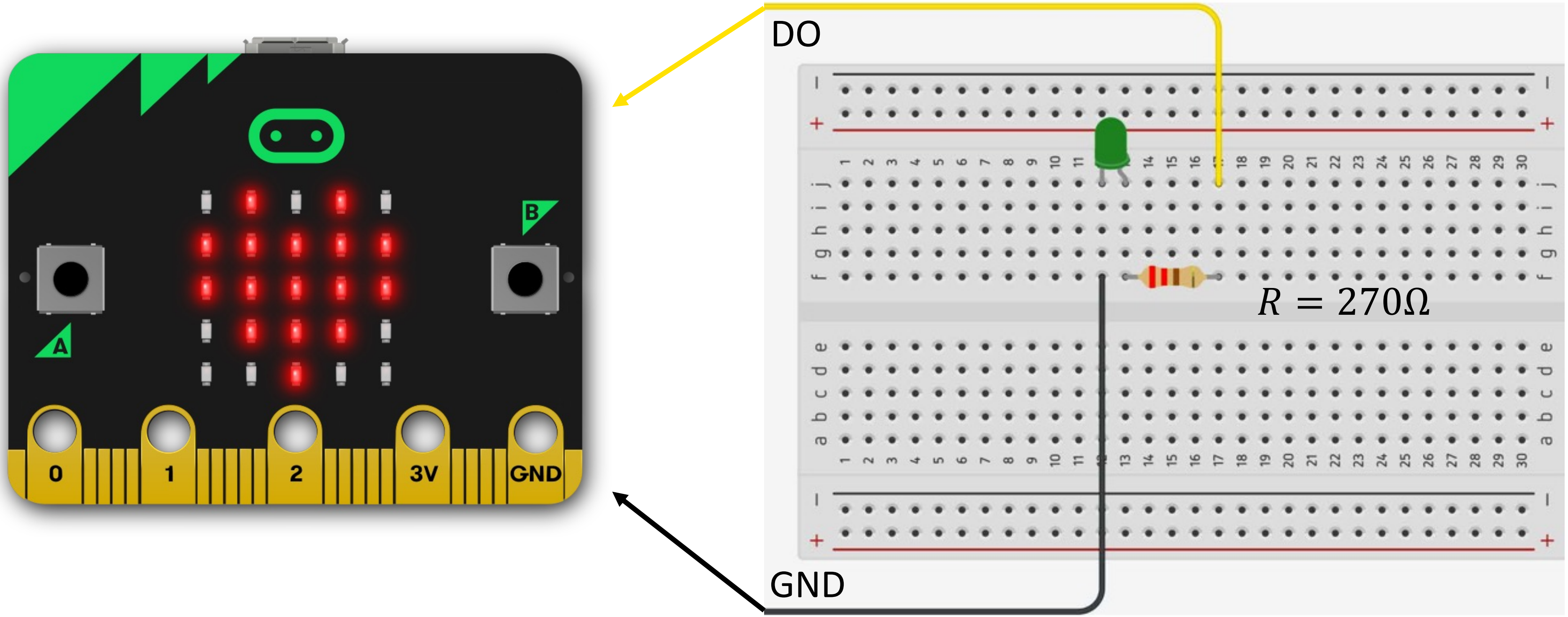
```python
from microbit import *

while True:
    adc = pin0.read_analog()
    volt = (3.3/1023)*adc
    degC = 100*volt - 50

    display.scroll(round(degC))
    print((round(degC, 1),))

    sleep(5000)
```

Temperature with Alarm

Hans-Petter Halvorsen

Table of Contents

LED Wiring

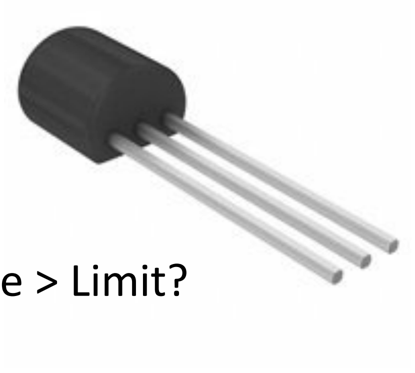


DO

GND

$R = 270\Omega$

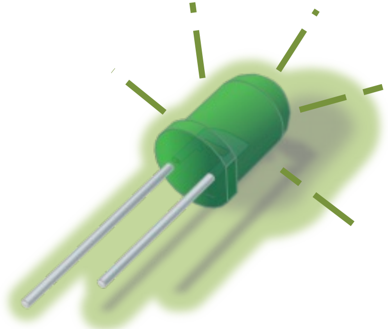
Python



Temperature > Limit?

No

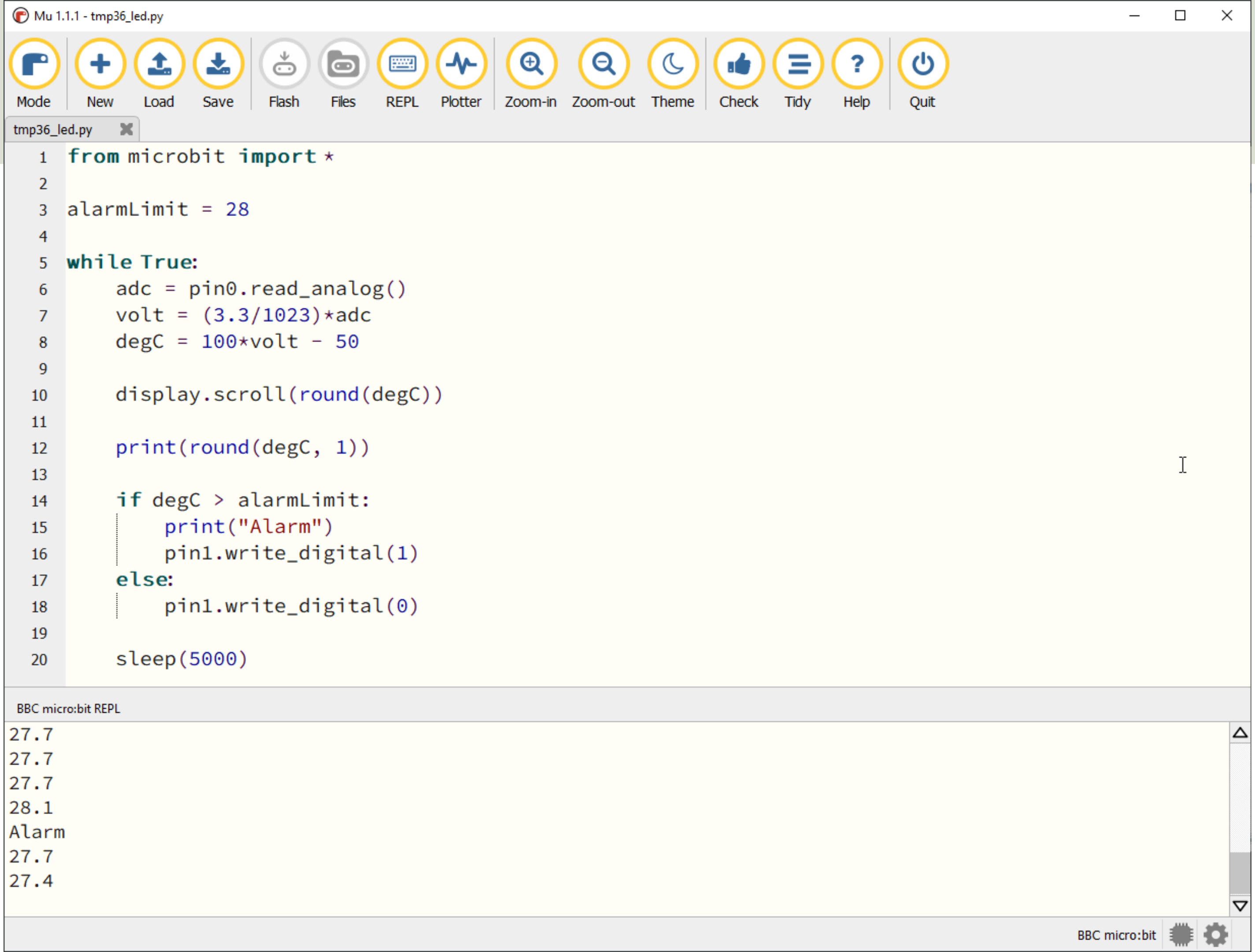
Yes



LED OFF

LED ON

```
from microbit import *

alarmLimit = 28

while True:
    adc = pin0.read_analog()
    volt = (3.3/1023)*adc
    degC = 100*volt - 50

    display.scroll(round(degC))

    print(round(degC, 1))

    if degC > alarmLimit:
        print("Alarm")
        pin1.write_digital(1)
    else:
        pin1.write_digital(0)

    sleep(5000)
```



```
from microbit import *

alarmLimit = 28

while True:
    adc = pin0.read_analog()
    volt = (3.3/1023)*adc
    degC = 100*volt - 50

    display.scroll(round(degC))

    print(round(degC, 1))

    if degC > alarmLimit:
        print("Alarm")
        pin1.write_digital(1)
    else:
        pin1.write_digital(0)

    sleep(5000)
```

```
27.7
27.7
27.7
28.1
Alarm
27.7
27.4
```

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: https://www.halvorsen.blog