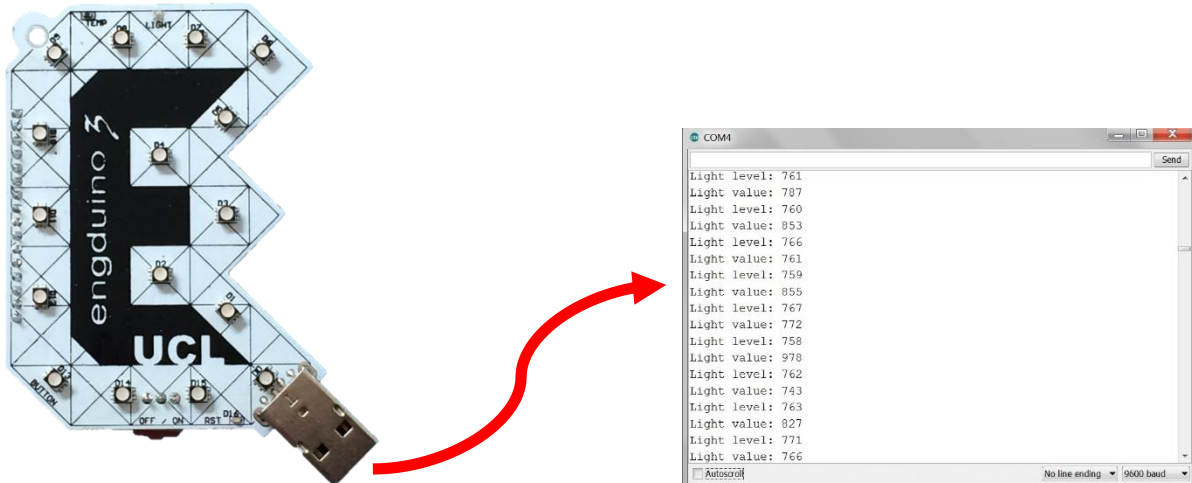


DATA VISUALISATION: PYTHON

It's useful to view sensor readings using the Serial Monitor by printing out the values using `Serial.print()` and `Serial.println()` functions. The Engduino sends the information to the serial interface which is actually the USB connector.



Everything that you print is output through the serial port.
In this case the serial port is the USB connector.

Sometimes it is easier to understand the meaning of data if we can plot values on a graph. There are several ways of doing this with the Engduino and in this worksheet we are going to explain how to do it using Python. Some special libraries are needed, you should install:

- [SciPy](#) which is a set of software for mathematics, science, and engineering.
- [drawnow](#) which provides a feature for redrawing a graph. You might need to install [Python pip](#) first.

At the top of your Python program you must import the following Python modules:

```
import serial
import numpy
import matplotlib.pyplot as plt
from drawnow import *
```

The Python serial module allows you to read from the serial port; everything which is printed to the Engduino's serial port can be used by your Python program. You have to create a serial object and to tell the Python program what the characteristics of the serial port are, like this:

```
arduinoData = serial.Serial('com4', 9600)
```

This says that the Engduino is plugged in to COM port number 4 and that we will communicate with it at a speed of 9600 bits per second. On a computer running Linux the port might be labelled `/dev/ACM0` or similar. The line speed is sometimes described as the baud rate.

Note: this speed must match the speed specified in the setup loop in your Engduino sketch. Your Engduino sketch should have this line in the `setup()` function:

```
Serial.begin(9600);
```

Now let's look at a simple Python 2.7 example that will plot the light values from the light sensor.

```
import serial
import numpy
import matplotlib.pyplot as plt
from drawnow import *

arduinoData = serial.Serial("/dev/ttyACM0", 9600)

lightReadings = []
count = 0

plt.ion() # use interactive mode to plot data

def doPlot():
    plt.ylim(0,1023) # Set limits for y axis.
    plt.title('Light level in classroom 4B at 9 am')
    plt.grid(True) # Draw a grid
    plt.ylabel('Light ') # Set label for y axis
    plt.plot(lightReadings, 'ro-', label='Light readings')
    plt.legend(loc='upper left') # Position of legend

while True:
    while (arduinoData.inWaiting() == 0): # Wait until there is data
        pass
    arduinoString = arduinoData.readline() # Read from the serial port
    temp = int(arduinoString) # Convert to an integer
    lightReadings.append(temp) # Add to the array
    drawnow(doPlot) # Update the graph

    count = count + 1 # Delete the first element
    if(count > 50): # so that the array remains
        tempF.pop(0) # no larger than 50 items
```

Let's take a closer look at the `'while True:'` loop. What this means is:

- a) Check to see if any data has been sent to the serial port. If no data has been received, go to the start of the loop again.
- b) Read a line of data into a string.
- c) Convert the string into a number (an integer) and store it in a variable called `temp`.
- d) Store the `temp` value at the end of the array called `lightReadings`.
- e) Re-draw the graph.
- f) If the array `lightReadings` is more than 50 elements long, remove the first element.
- g) Go back to step a).