LEDS

Now we're going to learn about to set up colors of the LEDs that will be displayed on your engduino
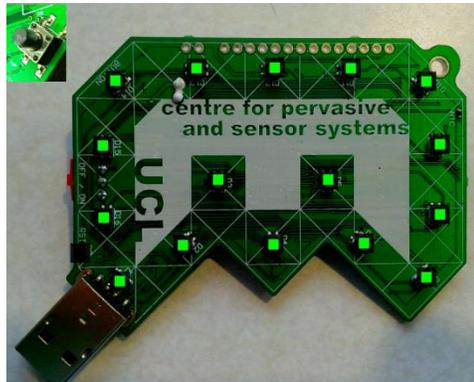
You can set the colour of all the LED's at the same time, by using 'set all LED'.

Here is an example where the colour is random, using TouchDevelop:

engduino→ set all LEDs(colors→ random)

To set the LEDs to a specific colour, instead of 'random', change this to the other colours available that are on TouchDevelop, which are *red, green, white, blue, orange, dark grey, grey, light grey and yellow*.

E.g. Setting all the LEDs to green.

engduino→ set all LEDs(colors→ green)

Set the colour to *black* to switch the LEDs OFF.

You can set the colour of a single LED, by using 'set LED(*LED number, colour*)'.

The number of the LED is labelled on the Engduino, from 1-16. Note: LED 0 is on the side of the Engduino.

Here is an example where LED 1 is set to red:

♻engduino→ set LED(1, colors→ red)

You can also use the 'colors → from rgb(r, g, b)' function to use other colours. Here is the link, where you can find the RGB values for different colours:
http://www.rapidtables.com/web/color/RGB_Color.htm

USING LOOPS

If you wanted to set the odd numbered LEDs to red and even numbered LEDs to green, then instead of listing out each one individually, you can use loops!

Here is how you would do this:

```
for 0 ≤ i < 17 do
    ♻engduino→ set LED(0, colors→ black)
    if math→ mod(i, 2) = 0 then
        ♻engduino→ set LED(i, colors→ green)
    else
        ♻engduino→ set LED(i, colors→ red)
    end if
end for
```

The 'for' loop always starts at 0, and it goes up to 17 so that it includes all 16 LEDs. We first set LED 0 to black, as we do not want this one to be on (it is the small LED on the side).

We use the 'math → mod' function and an 'if statement' to separate the odd even and odd numbers (the remainder of an even number divided by 2 would equal to 0, whereas odd number divided by 2 would give a remainder).

Here is the result:

centre for pervasive
and sensor systems