ENGDUINO LIBRARIES: THE LEDS

This is a quick guide to some of the things you can do with the LEDs. The idea is that you can use this information to experiment and put together something interesting that you created for yourselves rather than something we designed for you. Try stuff out – see what happens and see what you can make.

The LEDs can be set individually, or all together, and they can be set to different colours and brightnesses by mixing together different amounts of red, green and blue. There are 16 LEDs – and if you look on the board you'll see their numbers (D0 – D15) written in white next to them.

HEADER

At the top of your program you must have the following line – just like in the example

```
#include <EngduinoLEDs.h>
```

SETUP()

In the setup() function, you must have the following line – again, just like the example

```
EngduinoLEDs.begin();
```

BASIC FUNCTIONS

There are a number of functions that you can call to set the LEDs. The LEDs are numbered from 0 to 15 inclusive.

SET THE COLOUR OF ALL LEDS:

You can set the colour of all the LEDs at the same time, just as you did in the example you were given. So, to set all the LEDs to the colour BLUE, at maximum brightness, we use:

```
EngduinoLEDs.setAll(BLUE);
```

The full set of colours available to you is **RED, GREEN, BLUE, YELLOW, MAGENTA, CYAN, WHITE** and, although it's not strictly a colour, **OFF**.

CONTROLLING BRIGHTNESS

Maximum brightness is often a bit bright, so we can turn it down by selecting a number that ranges from 0 (= off) to 15 (=brightest). To set the LEDs to a fairly dim YELLOW, we use:

```
EngduinoLEDs.setAll(YELLOW, 2);
```

SETTING A SINGLE LED

If we want to set the colour of a single LED, we can do that too. For example to set LED0 (which is just by the USB connector) to be RED at maximum brightness we type:

```
EngduinoLEDs.setLED(0, RED);
```

But you can also control the brightness – to set LED2 to GREEN at brightness 3, then we use:

```
EngduinoLEDs.setLED(2, GREEN, 3);
```

Try this - you should see that LED2 is a lot less bright than LED1. Brightness can vary between 0 (off) and 15 (brightest) - experiment....

MORE ADVANCED FUNCTIONS

The primary colours of light are red (R), green (G), and blue (B). If we mix these in different ways, we can get all sorts of shades of light. The LED libraries allow users to set LEDs by the specific mix of brightnesses of R, G, and B. Remembering that the brightness lies between 0 and 15:

Set all LEDs with red brightness = 2, green = 2; and blue = 0

```
EngduinoLEDs.setAll(2, 2, 0);
```

Red and green mixed together in equal proportions make yellow – so this is yellow and, because we chose low numbers, it's not very bright. But what if we want to make orange – which isn't on our original list? Well, orange is like yellow, but with a bit more red in it, so let's make the red a bit brighter and keep the green the same:

```
EngduinoLEDs.setAll(5, 2, 0);
```

Voila, orange.

We can do the same thing for individual LEDs. So, to set LED4 to a salmon pink (red = 9, green = 6, blue = 2), we might use:

```
EngduinoLEDs.setLED(4, 9, 6, 2);
```

There are even more ways of setting the LEDs for advanced programmers. You can find these on our website http://www.engduino.org/doxygen/ but you need to know quite a lot to understand this.

IDEAS FOR PROJECTS WITH THE LEDS

- Make a flashing rear cycle light.
- Make an electronic candle where the top flickers, or any type of Christmas tree decoration.
- Make the LEDs spell out the letters UCL one after another.
- Make a single LED seem to run round the board.
- Make the LEDs fade in and out
- …. It's up to you.