Write Your Light Sketch

Engduino®

You are now ready to write your first program! First, you need to open a new sketch using the Arduino menu File=>New.

This gives you the two main elements in Arduino C. The "Setup" and "Loop" routines. Type out the code as follow:

Blink

```
#include "EngduinoLEDs.h"
void setup  ( )   {
     // put your setup code here, to run once:
     EngduinoLEDs.begin();
}
// the loop function runs over and over again forever
void loop ( ) {
     EngduinoLEDs.setAll(RED);
     delay(1000);
     EngduinoLEDs.setAll(OFF);
     delay(1000);              // wait for a second
}
```

Traps:

There are some common mistakes that result in errors when you type out the codes:

- Capital/ small letters
- Semi-colons at the end of each statement
- Curly brackets - remember to open and close the curly brackets to indicate the start and finish of the codes

The elements of an Arduino Sketch

1 Header

```
#include "EngduinoLEDs.h"
```

The library "EngduinoLEDs", and other libraries are codes that other people have written to help to make programming easier for the users. You can include these pre-written codes by including the header files. In this case, we would like to use the 16 multi-color LEDs, and hence, we include the "EngduinoLEDs.h" header file.

2 Setup

```
void setup ( ) {
     EngduinoLEDs.begin();
}
```

This setup function contains codes that only runs at the setup stage of the program. It is only run once; then the program moves on to the loop function, or the main program.

In our example, we call "EngduinoLEDs.begin();". This will make the program runs the codes that are required to prepare the LEDs for further programming.

3 Loop

Void loop() {}

The loop function, which is the main program, contains the codes that are run continuously. The code runs from top to bottom inside the loop function, and then start from the first line again. The curly brackets define the start and the finish of a function.

In our example, first we set all the lights to red (be careful of the capital letters), wait for 1 second (1000 microseconds), and then turn all the lights off, then wait for another second before repeating the code.

Individual Lights

Engduino®

LED stands for Light Emitting Diode - it is a cheap and low energy way semiconductor light source. On the Engduino, there are multicolour 16 LEDs. Sometimes we call these RGB (Red, Green and Blue) LEDS. You can mix different amount of RGB to get any shade of colour.

From the blink sketch, you can try different colours lights and delay time to make different patterns of LEDs. The pre-set colours available are

LEDCo

```
#include "EngduinoLEDs.h"
void setup ( ) {
    // put your setup code here, to run once:
    EngduinoLEDs.begin();
}
// the loop function runs over and over again forever
void loop ( ) {
    EngduinoLEDs.setLED(0, RED);
    EngduinoLEDs.setLED(1, GREEN);
    EngduinoLEDs.setLED(2, BLUE);
    EngduinoLEDs.setLED(3, YELLOW);
    EngduinoLEDs.setLED(4, GREEN);
    EngduinoLEDs.setLED(5, MAGENTA);
}
```

Try other pre-defined colours:

RED
GREEN
BLUE
YELLOW
MAGENTA
CYAN
WHITE
OFF

Programming Traps:

There are some common mistakes that result in errors when you type out the codes:

- Capital/ small letters
- Semi-colons at the end of each statement