Jewellery Making

Computer style

Introducing Engduino

Engduino is a new piece of technology from University College London.

It's a wearable so you can use your imagination to program its LEDs and wear it as a piece of jewellery.

You connect it to your PC or Mac – or even to your Linux machine – and program it. Then you disconnect it and wear it!

You can start learning to program it from scratch. You can help your little sister or brother change your programs to make their own personal jewellery. Here are some teenagers wearing the programs they wrote.

And here is a Year 4 student wearing a program she edited. She started with a ready-made program and changed it to flash the colours she chose.

Stay safe!

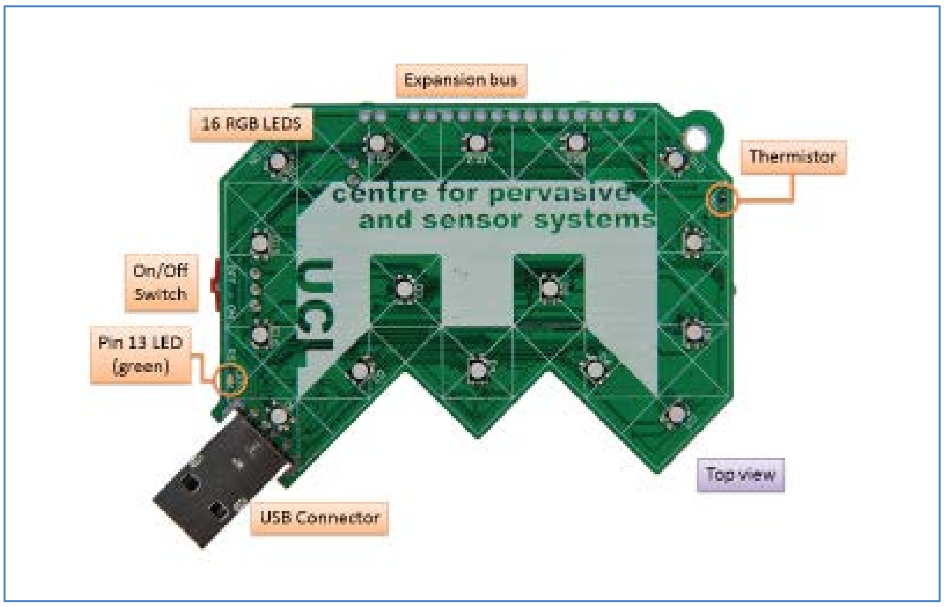
Don't put the Engduino in your pocket or in a bag with keys or coins in case you short out the battery.

Take off any other jewellery before you wear it.

A Closer Look

Engduino has 16 LEDs.

It also has a temperature sensor, a motion sensor and an IR transmitter/ receiver. We'll learn about those later.



You connect it to a computer using its USB connector. It works without being connected to a computer because it has a battery that gets charged when it is connected.

All the LEDs have numbers. You can make them

- all light up in the same colour at the same time
- all light up at the same time but in different colours
- light up one by one
- turn off all at the same time or one by one.

By switching them on and off we can make the Engduino into a piece of unique flashing jewellery.

Some Planning

Take a planning sheet or download one from the network – ask an adult where it has been saved.

The planning sheet shows the LEDs as small circles with numbers.

You will need some felt pens to colour in the LEDs on the sheet to show the colours you want to use.

The Arduino language recognises these colours – RED, GREEN, BLUE, YELLOW, MAGENTA, CYAN, WHITE – and OFF, although of course OFF is not really a colour.

Later, you will learn how to make more colouirs.

All LEDs Flashing in Different Colours at the Same Time

Use your planning sheet to decide which colours to flash the LEDs. With seven colours – or eight if you count OFF – some LEDs will flash the same colour. You don't have to use all the colours.

Using the Engduino for the first time is quite exciting. Choose some colours to make an exciting display.

Think about whether you want it to flash quickly or slowly and make a note on your planning sheet. Remember to make it lively and exciting.

Coding

We call an Arduino program a “sketch”.

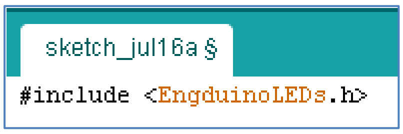
Our programs will have three sections.

First Section

First, we include the special code for the parts of the Engduino we want to use. You’ve probably used Python, so you recognise code like

import turtle or import time.

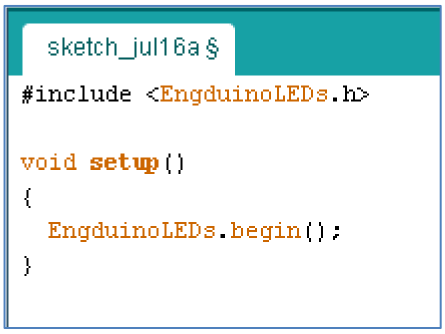
In Arduino, we use



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

Second Section

Next, we do some setting up. The setup section runs when we upload our program to the Engduino or restart it.



```
#include <EngduinoLEDs.h>

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

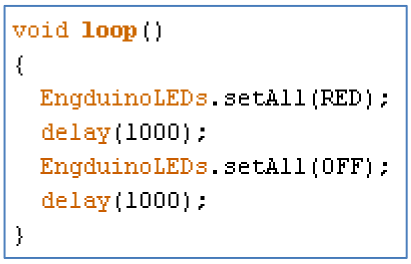
All we are doing here is starting the code we included.

Notice that the code to start the program sits inside curly braces.

You will soon learn that lines of code end in a semi colon.

Third Section

The third section is a loop that keeps on running our code for the Engduino. This means that it will keep on flashing until we turn it off or until the battery runs out.



```
void loop()
{
  EngduinoLEDs.setAll(RED);
  delay(1000);
  EngduinoLEDs.setAll(OFF);
  delay(1000);
}
```

What do you think this code will do?

Your Own Code

Make sure that the Engduino is connected to a USB port.

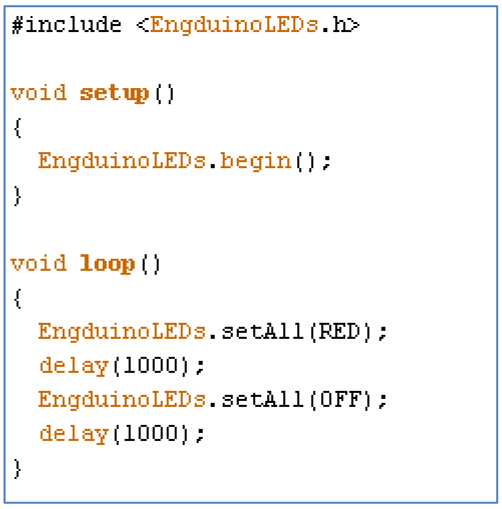
The amber light will come on to show that it is charging (unless it is already fully charged, of course).

Make sure that it is switched on – slide the red slider away from the amber light. If it has already been programmed, it will start to run its program. It will probably flash some LEDs.

Start the Arduino IDE by double clicking on the Arduino icon and choosing Run.

Now write your code.

Here is some sample code:

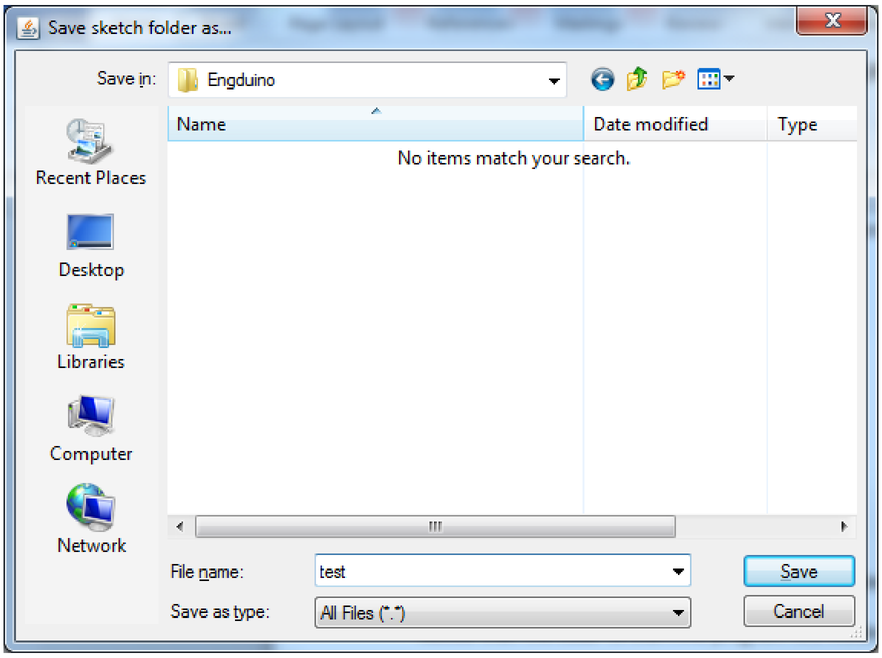


```
#include <EngduinoLEDs.h>

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

void loop()
{
  EngduinoLEDs.setAll(RED);
  delay(1000);
  EngduinoLEDs.setAll(OFF);
  delay(1000);
}
```

Choose File → Save As

Go to your home area and create a folder called Engduino.

Now give your sketch a name. I called mine test.



Arduino will make a folder called test. Inside it, Arduino will save the test sketch as test.ino

Next, click on the tick icon to compile the sketch. Compiling means turning it into special code that the Engduino can run.

If you have not made any errors, you will see a message stating that the Arduino IDE has "done compiling" – it's finished. Click the arrow icon to upload (send) your compiled code to the Engduino.

It should run your program.

The Code You Need to Know

delay(nnnn)	sets a delay in thousandths of a second.
delay(1000)	makes the code pause for a second before it executes the next instruction you gave it
Engduino.setAll(COLOUR);	sets all the LEDs on with the same colour. Don't enter colour, choose one of the colours from RED, GREEN, BLUE, YELLOW, MAGENTA, CYAN or WHITE.
Engduino.setLED(1,YELLOW)	Sets LED number 1 to Yellow. You can work out how to set other LEDs on to different colours
Engduino.setAll(OFF)	
Engduino.setLED(1, OFF)	

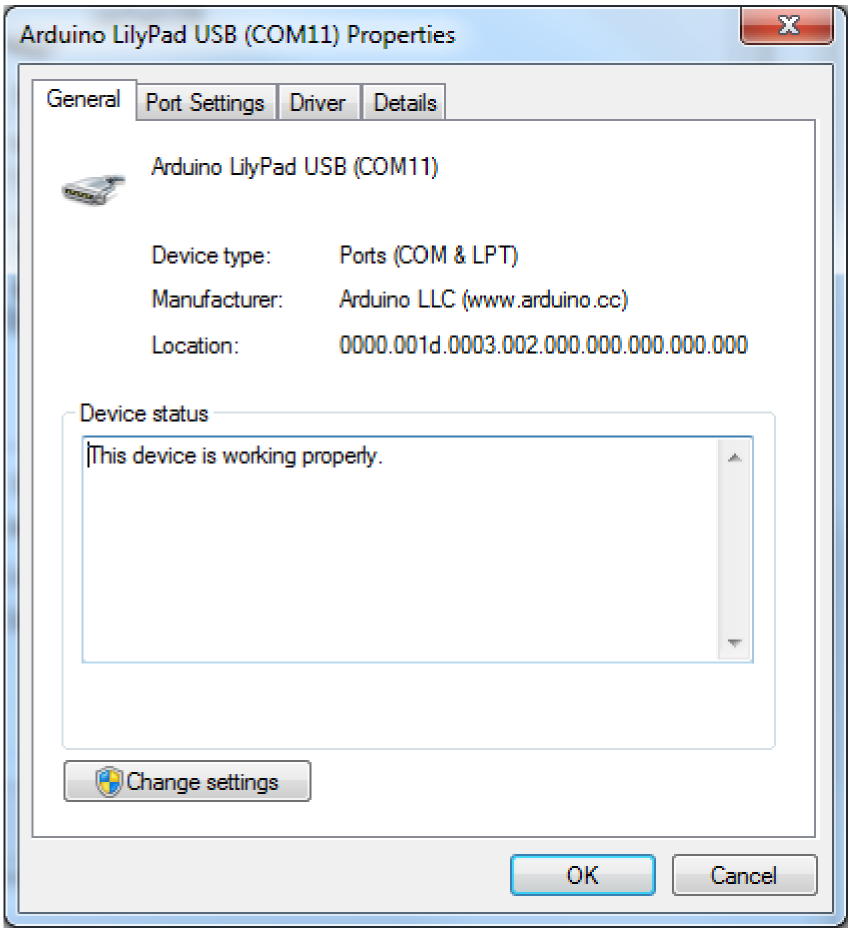
About Loops

In a loop, the Engduino runs each instruction, reaches the last one, then starts all over again. It keeps running until you switch it off or the battery runs out.

Troubleshooting - Hardware

The first time you use the Engduino, an adult will probably have set it up for you to make sure it works.

If you are working on your own, first make sure that you can see the Engduino when you choose Devices and Printers (in Windows 7). It will appear as LilyPadUSB if it is installed and switched on. Right click and choose Properties, then Hardware, then Properties.



Arduino LilyPad USB (COM11) Properties

General | Port Settings | Driver | Details

Arduino LilyPad USB (COM11)

Device type:	Ports (COM & LPT)
Manufacturer:	Arduino LLC (www.arduino.cc)
Location:	0000.001d.0003.002.000.000.000.000.000

Device status

This device is working properly.

The Engduino (also known as Arduino LilyPad USB) is installed on COM Port 11.

Back in the Arduino IDE, choose

- Tools → Board to check that the software is using the LilyPad Arduino USB
- Tools → Serial Port to check that it is using the correct COM (serial) port.

Troubleshooting – Software

If your program won’t compile

Look for typos, upper case instead of lower case, lower case instead of upper case, missing colons….

What do you think will happen with this code?

```
void loop()
{
  EngduinoLEDs.SetAll(RED)
}
```

It won’t compile

- because SetAll should be setAll
- and beause the colon is missing from the end of the line.

If your program compiles but doesn’t do what you expect

You probably have a logic error, which means that the computer is doing what you asked but not what you meant. You did not think it through carefully! Don’t worry, everyone has made logic errors at some time, it is normal. You learn by making mistakes.

Suppose you want the Engduino to flash red at one second intervals.

Why does this code not flash the LEDs?

```
void loop()
{
  EngduinoLEDs.setAll(RED);
  delay(1000);
  EngduinoLEDs.setAll(OFF);
}
```

If you can’t see why, discuss it with a friend. Try it out. Get it to work!