

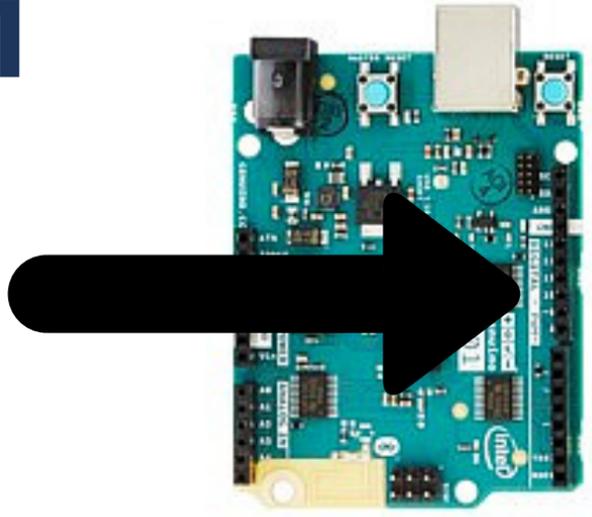
Working with Arduino Microcontrollers such as the **Genuino 101* board** requires an understanding of the Arduino code underpinning the project. The Arduino coding language is an open source software package that allows you to communicate with the Arduino Microcontroller. We upload our written programs to our Arduino Microcontroller by connecting it to the computer via a USB lead. Let's look at the basics of Arduino and Microcontrollers.

Knowing your 101

Taking a look at a Microcontroller such as the Genuino 101 board you will see there are a number of input or output ports called pins. On the 101, as with most Arduinos, they are numbered. In this case from 1 to 13.

In addition you will notice other ports including one labelled GND or ground. The ports are numbered so that we can program the inputs and outputs we connect to each of these ports, but more on that later.

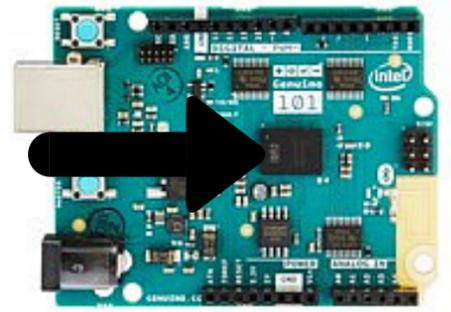
STEP 01



STEP 02

Open Arduino

If you have not already installed the Arduino Software, access the getting started guide here <http://bit.ly/genuinsetup>. Otherwise open Arduino and then click on **File >> Examples >> Basics >> Blink** to open the most basic of Arduino code.



STEP 03

Naming components

Here we are naming the output pin 13 (remember all the pins are numbered) 'led' just to make things easy. This way when writing and troubleshooting code we are looking not for 13 but led. When you work with motors, servos etc, you can name these as such too, to make your and your students lives easy.

STEP 04

Setup the board

Physical computers, such as the 101, "Sense" input then decide what to "Do" by processing, resulting in an output or "Reaction".

This section of code is just letting the board know whether it needs to be ready to sense or react. In this case we have connected an led, or an output, to point 13 so we call it an output. If we were planning to connect a light or sound sensor, we would need to call it an input.

STEP 05

Loop

The loop section of code is where we tell the Arduino what we want it to do, and as the name suggests the program is repeated continuously. The first line `digitalwrite(led, HIGH)` is telling the microcontroller to turn up the voltage to output, or port, 13 to high, in other words turn the power on. This will turn on our led. In the second line, 'delay' is basically saying 'wait'. It is represented in milliseconds, so in this case wait for 1 second (1000ms = 1s love maths at work) before doing anything else. The second last line of code `digitalwrite(led, LOW)` is telling the microcontroller to turn down the voltage to output 13, turning the led off. Don't forget because this is a loop it will be repeated, making the led blink, hence the name of the program!

```

Blink | Arduino 1.5.3-Intel.LU.4
File Edit Sketch Tools Help
Blink
/*
  Blink
  Turns on an LED on for one second, then off for

  This example code is in the public domain.
  */

// Pin 13 has an LED connected on most Arduino bo
// give it a name:
int led = 13;
  
```

What does the code do?

This code, as the name suggest makes an led blink or flash. To understand the code we will take a look at the 3 main sections: Naming; Setup; Loop (Program section.)

```

// Pin 13 has an LED con
// give it a name:
int led = 13;
  
```

```

/ the setup routine run
void setup() {
  // initialize the digi
  pinMode(led, OUTPUT);
  
```

```

void loop() {
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);
}
  
```