Prerequisites

This worksheet builds upon the Lab 1 worksheet. You will have needed to complete the Lab 1 worksheet before you can continue with this lab. Out of your Arduino pack, you will need: 1 Arduino, 1 breadboard, 1 LED, 1 220Ω resistor, and 1 jumper cable.

Objectives

The objective of today's lab is to learn how to use an electronic breadboard to create a circuit and to control an external output on the Arduino.

We will be taking the scenario from the previous worksheet and changing it to communicate with an external LED. Whilst this may sound complex, we are able to use the code from the previous worksheet as a starting point; therefore, it is only one new line of code and changing three existing lines of code.

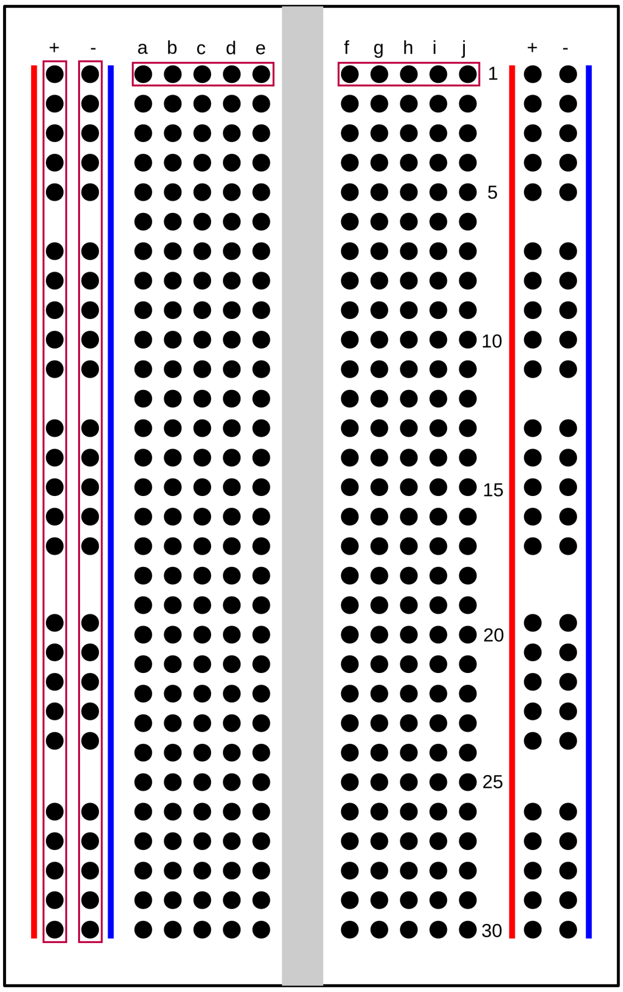
Understanding a Breadboard

An electronic breadboard is a useful tool to create quick circuits. It becomes particularly helpful in our use case, as we can place our Arduino within the breadboard without any further adapters.

A breadboard can be thought of as a collection of rows ($1 \rightarrow N$) and columns (A→J). Each row is wired together into two groups — divided by the centre separator. The first group is the columns A→E (in our example included in Figure 1), and the second group is columns F→J. Each row is numbered from $1 \rightarrow N$. Any item plugged into a pin will be accessible by any other pins within that pins row group.

For instance, if you plugged the Arduino output for Pin D2 into column H on row 13, you would then be able to plug in an LED leg into column J on row 13. This would create a complete circuit — given that you have also completed the part of the circuit for the other LED leg.

There is also four power rails at either side of the breadboard — two on the left, and two on the right. The rails on either side are traditionally used for positive and negative power. In our case this would be used for +5v and GND. All the rows within a rails column are considered connected together. Figure 1 depicts the two power rail groups on the left side of the breadboard, as well as depicting the two row groups for row 1.



Original image source: `https://freesvg.org/breadboard`

Figure 1: An example electronic breadboard. There are four groups highlighted. The positive and negative column groups, and the two groups that are wired together for each row.

1 Creating the circuit

The circuit needed for this worksheet is: Arduino Pin D2 →220Ω resistor →LED →Arduino GND.

To do this, place your Arduino into the breadboard. Once this is done, use the pin out diagram linked in lecture 1 to find which pin on the Arduino is Pin D2. You then need to connect one leg of the 220Ω resistor to Pin D2 of your Arduino through your breadboard. Then connect the other leg of the 220Ω resistor — mind how the rows and columns are connected, ensure the resistor isn't bypassed in the circuit — to the positive leg of the LED (the longer leg). Finally, connect the negative leg of the LED to the ground PIN on your Arduino through the jumper cable. You won't see anything visual happen at this stage, you need to complete the code tweaks below before the LED will light up.

2 Changing the existing code

Firstly, open your sketch from the previous worksheet into Arduino IDE. Once this is open, save the sketch under a new name (‘File’→‘Save As’). This will make sure you still have a copy of the sketch from Lab 1.

The first thing we are going to do is create a new definition for a ‘*LED_PIN*’ which will map to the Arduino Pin D2. To do this, copy the code below at the top of your file (before the Morse code definitions).

```
#define LED_PIN 2
```

The internal LED on an Arduino is referenced in exactly the same way as you would reference an external pin. This means all we need to do is change anywhere within our code that uses ‘*LED_BUILTIN*’, to use the newly defined external pin ‘*LED_PIN*’. This means changing the code to the following lines:

```
// this line is in the setup function
pinMode(LED_PIN, OUTPUT);


// these two lines are in the main function
// note: there is a line of code between these two lines
digitalWrite(LED_PIN, HIGH);
digitalWrite(LED_PIN, LOW);
```

Try it out on your Arduino

You now should be able to compile your sketch (‘Sketch’→‘Verify/Compile’) and then upload to your Arduino (‘Sketch’→‘Upload’). You should see that your Arduino now blinks the externally connected LED in the same pattern as it did on your internal LED.

If you have managed to get the external LED to turn on and off with the Morse code signal, see if you can change your code so that ‘dot’ signals are displayed on the internal LED, and ‘dash’ signals are displayed on the external LED.

Help

If you are having issue compiling the project or don't understand where each section goes, you can see a complete version of the sketch with additional source code comments at `https://github.com/dsbrennan/mac-233-arduino-labs/blob/main/lab-2-led/lab-2-led.ino`.