Lab 3: Fading Lab

Objectives

This lab introduces *for* loops and *while* loops through control of LEDs, piezo buzzers, and DC motors. Students will observe the how the two loops work and how to perform the same actions with the two loops.

Materials

1) Arduino Uno
2) MakeBlock Shield
3) 1 × LED
4) 1 × 220 Ω Resistor
5) 1 × DC motor
6) 1 × buzzer
7) Wires for Building Circuits
8) Wire Cutters
9) Wire Strippers

Theory

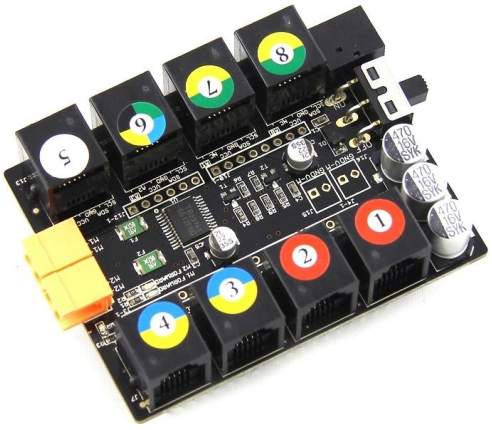

Figure 1: Me Base Shield

The **MakeBlock Me Base Shield**, shown in Figure 1, is a device which stacks on top of the Arduino. This device provides simple phone cord connections (RJ25) between ports and various modules. Modules can include various MakeBlock sensors, motor drivers, and lights. Figure 2, shows the different ports on the shield.

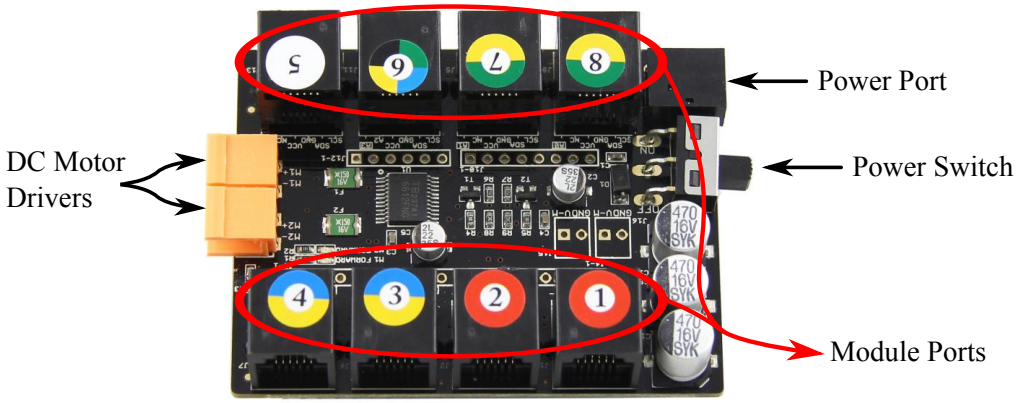

Figure 2: Shield Diagram

Tag Color	Function	Typical Me Modules
●	Red represents the output voltage of Vin (9-12 VDC), which is the power input voltage. Generally, it is connected to some voltage driver module with 9-12V.	Motor Driver, Servo Driver, Microstep Driver, Lego Motor Bridge
●	Orange, blue, green, black and white represent the output voltage of Vcc (5VDC), which is the Stable output voltage. Generally, it is connected to some voltage driver module with 5V.	Ultrasonic Sensor, Limit Switch, RJ25 Adaper
●		Infrared Receiver , Line Finder, Bluetooth, RJ25 Adaper
●		RJ25 Adaper
●		RJ25 Adaper
○		RJ25 Adaper

Figure 3: Module Color Code

PORT NO.	Tag Color	Compatible Module types	Typical Me Modules
1	● ○	(9-12VDC) driven modules	Motor Driver, Servo Driver, Lego Motor Bridge
2	● ○	(9-12VDC) driven modules	Motor Driver, Servo Driver, Lego Motor Bridge
3	● ● ○	1 or 2 way digital modules or I2C modules	Line Finder, Bluetooth, Infrared Receiver, Lego Digital Bridge, RJ25 Adaper
4	● ● ○	1 or 2 way digital modules or I2C modules	Line Finder, Bluetooth, Infrared Receiver, Lego Digital Bridge, RJ25 Adaper
5	○	I2C modules	I2C modules, RJ25 Adaper
6	● ● ● ●	1 or 2 way digital or analog modules or I2C modules	Line Finder, Bluetooth, Infrared Receiver, Lego Digital Bridge, Lego Analog Bridge, RJ25 Adaper
7	● ● ○	1 way digital or analog modules or I2C modules	Infrared Receiver, Lego Analog Bridge, RJ25 Adaper
8	● ● ○	1 way digital or analog modules or I2C modules	Infrared Receiver, Lego Analog Bridge, RJ25 Adaper

Figure 4: Module Port Legend

while Loops

A **while** loop repeats a set of code until a condition is true. Below are some examples that change the variable x from 0 to 100 in steps of one.

```
1 // Method 1
2 int x = 0;
3 while (x < 100)
4 {
5   x = x + 1;
6 }
7 // Method 2
8 x = 0;
9 while (x < 100)
10 {
11   x +=1;
12 }
13 // Method 3
14 x = 0;
15 while (x < 100)
16 {
17   x++;
18 }
```

do-while Loops

While loops test a condition before executing the code in the loop. The code in the loop will never execute if the condition is never true. Consider the following code:

```
1 int x = 90;
2 do
3 {
4   x++;
5 } while (x < 40);
```

A while loop would not execute the code within the loop because the condition is not met. However, a **do-while** loop will always execute the code within the loop at least once.

for Loops

When dealing with while loops, programmers tend to set up counter variable to determine whether a condition is met (x in the previous examples). A **for** loop is a powerful loop that combines the following steps:

- Initialization of a counter variable
- Conditional test
- The change of the counter

The following code cycles 13 times and adds the counter variable and a constant to the variable y.

```
1 int y;
2 int const = 6;
3 for (int x = 1; x <= 13; x++)
```

```
4 {
5   y = x + const;
6 }
```

Laboratory Exercises

1. Check the resistor and be sure you picked the right one. Ask a TA if you have a problem finding the value of the resistor.
2. Use the jumper wires and make the circuit in figure below. (DO NOT CONNECT ANY THING TO THE USB PORT OR POWER SOURCE BEFORE CHECKING WITH TAs). Supplying too much power and wrong connections can damage the Arduino.

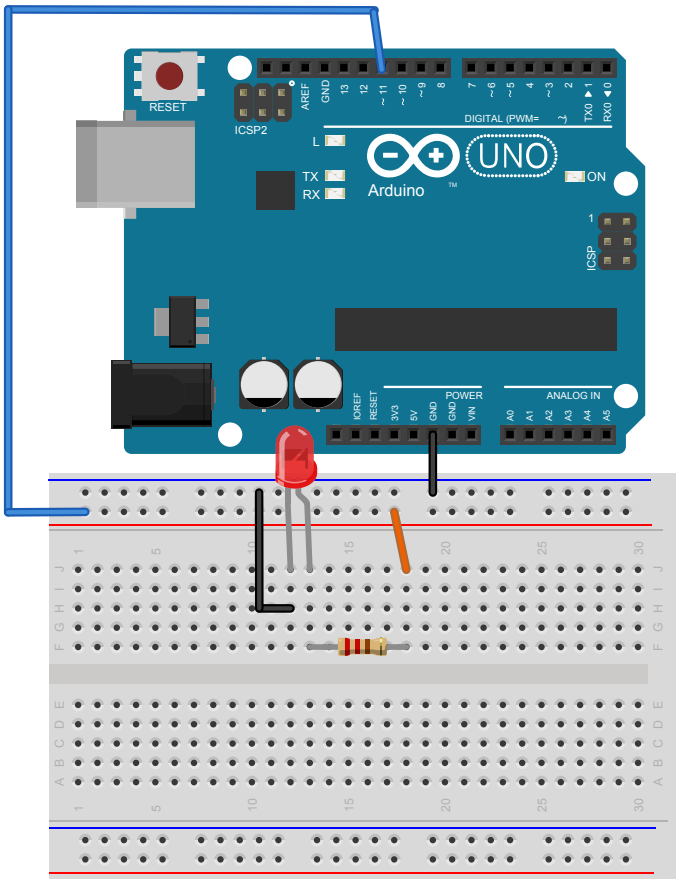

Figure 5: Fade circuit

3. Now open the sketchpad and check the setting for port number and board type.
4. In sketchpad, try to make a new code for making a fading LED. You can use the code below. *LED* is the pin number and the *brightness* is a number from 0 to 255. For this part, make a **for** loop to incrementally increase and decrease the brightness. Ask a TA if you have questions setting up a **for** loop.

```
1 analogWrite(LED, brightness);
```

5. Try to use the volt meter to see how the voltage changes with different brightness values. Put the positive probe on the output of port 11, and put the negative probe on ground.

6. Unassemble the circuit and put every thing back in the boxes.

Making a Buzzer

1. Use the jumper wires and make the circuit in figure below. (DO NOT CONNECT ANY THING TO THE USB PORT OR POWER SOURCE BEFORE CHECKING WITH TAs). Supplying too much power and wrong connections can damage the Arduino.

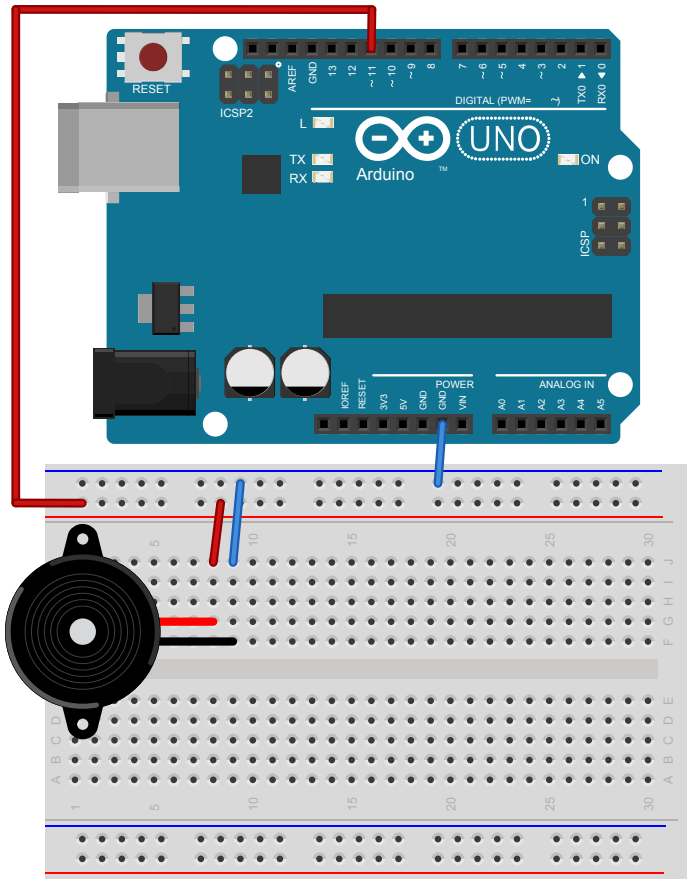

Figure 6: Buzzer circuit

2. In sketchpad, try to make a new code for making a buzzer produce sound. You can use the code below which *Buzzer* is the pin number and the *buzzVal* is a number from 0 to 255. For this part, make a *while* loop to incrementally increase and decrease the tone. Ask a TA if you have questions setting up a **while** loop.

```
1 analogWrite(Buzzer, buzzVal);
```

3. Unassemble the circuit and put every thing back in the boxes.

Move a DC motor

1. Wait for TAs to show you how to connect the MakeBlock shield to the Arduino. Connect a DC motor to the shield.

2. Use the jumper wires and make the circuit in figure below. (DO NOT CONNECT ANY THING TO THE USB PORT OR POWER SOURCE BEFORE CHECKING WITH TAs). Supplying too much power and wrong connections can damage the Arduino.

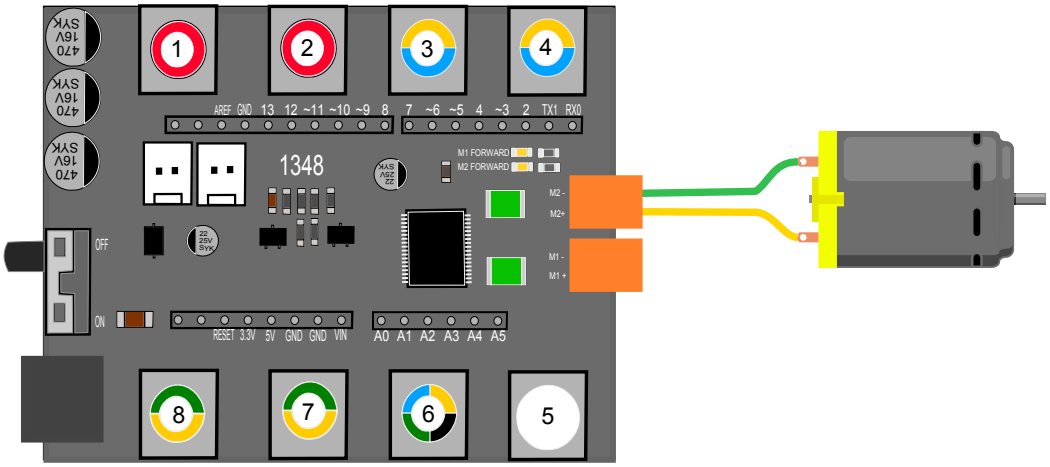

Figure 7: DC Motor circuit

3. In sketchpad, try to make a new code a which makes a DC motor move at different speeds. You can use the code below where *speed* is a number that ranges from -255 to 255. For this part, make a **do-while** loop to incrementally increase and decrease the speed. Ask a TA if you have questions setting up a **do-while** loop.

```
1 MeDCMotor motor(M1); // Initalizes DC motor on port M1
2 motor.runSpeed(speed); // goes in the do-while loop
```

4. Unassemble the circuit and put every thing back in the boxes.

5. Do not leave the class until you understand every step that you did (you'll need these skills later on in the year).