Lab 4: Functions Lab

Objectives

This lab introduces functions to students. This lab explains the inputs and outputs of functions. Students will make their own functions which will control various LED patterns. Student will learn the difference between *global* and *local* variables.

Materials

1) Arduino Uno
2) MakeBlock Shield
3) MakeBlock Servo
4) MakeBlock Servo Driver Module
5) 1 × RJ25 Cable
6) 3 × LEDs
7) 3 × 220 Ω Resistors
8) Wires for Building Circuits
9) Wire Cutters
10) Wire Strippers

Theory

What Is a Function?

A *function* is a section of code that can be called, perform an action, and return a value. Every C++ or sketchpad code has at least one function. The main(), or loop() in sketchpad, function is the function that required for each program. Other functions can be called within the main function.

Every function has a name which is used to call the function elsewhere. A function executes the internal code until it reaches a *return* statement or the last line of the function.

Before you create a function, you need to understand how to declare a function. There are three necessary steps in declaring a function:

- return type (eg: int, void, bool, etc.)
- the name (calcVolume, LedPattern, etc.)
- the number and type of input parameters

For example, consider the code below:

```
1 int calcVolume (int length, int width, int height);
```

The code above has an *int* variable return type, has the title *calcVolume*, and has three *int* variable inputs. A function can return any variable type. However, a *void* function does not need to return variable. Below is an example of the calculate volume function.

```
1 int calcVolume (int l, int w, int h)
2 {
3   int V = l * w * h;
4   return V;
5 }
```

An example of a full code is shown below:

```
1 // This code calculates the volume of cuboid
2 int calcVolume (int length, int width, int height);
3
4 void setup()
5 {
6   // no setup needed for this code
7 }
8
9 void loop()
10 {
11   int length = 5;
12   int width = 2;
13   int height = 4;
14   int Volume;
15
16   Volume =  calcVolume (length, width, height);
17 }
18
19 int calcVolume (int l, int w, int h)
20 {
21   int V = l * w * h;
22   return V;
23 }
```

The code above initializes the function *calcVolume* on line 2. This is required because the function is written below the main loop. However, line 2 is not necessary the *calcVolume* function (lines 19 to 23) is moved above the main loop.

Global vs. Local Variables

The code above uses *local* variables within the main and calcVolume loops. *Local* variables are declared within a specific loop and exists only within that loop. On the other hand, *global* variables are declared at the top of the program and can be called and modified in all loops. The code below shows an example of a code that utilizes both *local* and *global* variable types.

```
1 int gVar = 5; // global variable
2 int GenFunction (int locVar);
3
4 void loop()
5 {
6   int locVar = 7; // local variable
7   GenFunction (locVar);
```

```
 8 }
 9
10 void GenFunction (int lV)
11 {
12   gVar += lV;
13 }
```

The code above modifies the *global* variable *gVar*. The *GenFunction* has access to the global variable as previously described. However, the *GenFunction* does not know about the local variable *locVar* but does know the value of the variable *lV*. The *genFunction* indirectly knows the value of *locVar* because *locVar* is equal to *lV*.

Portion of Code	gVar
Initialization	5
1^{st} time through main loop	12
2^{nd} time through main loop	19
⋮	⋮

Laboratory Exercises

IR Remote Controller

1. Try to make a function with this structure to vary each LED shown in Figure 1:
   void output = pattLED(LED number, number of blinks, Frequency);
   Ask a TA if you need any help.

2. Now try to find a better use of this function for your project. Discuss about it with your group and a TA.

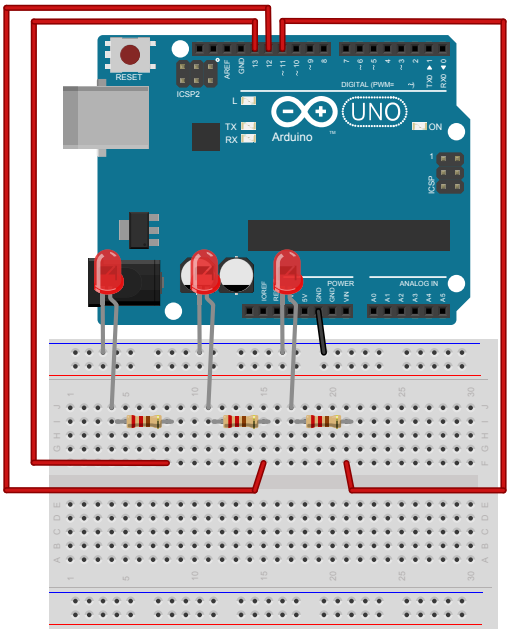

Figure 1: Multiple LED

Servo Motors

1. Attach the shield to the Arduino. Connect a servo to the shield via servo driver. In sketch pad, go to Makeblock examples and open the TestServoDriver example. Upload it to the Arduino.

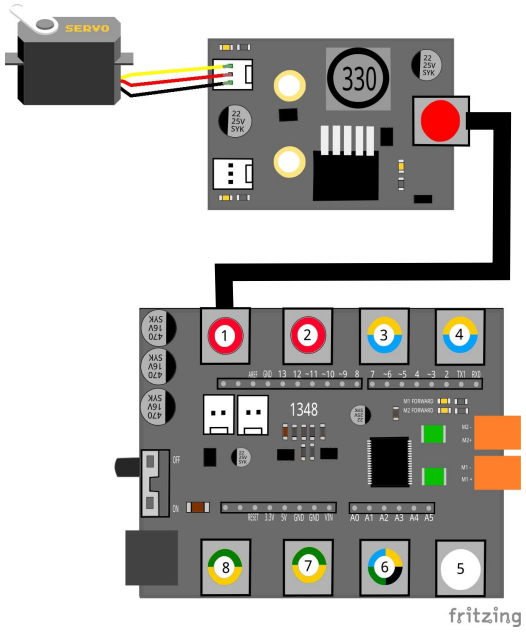

Figure 2: Servo Circuit

2. Now open a new file. Try to make a function to send the Servo to multiple angles (in degrees). Make the servo move to the following degree sequence: 0, 90, 45, 135, 90, 180, and 0.
3. Be sure you understand the difference between the DC motor and the Servo motor.
4. Unassemble the circuit and put every thing back in the boxes.
5. Do not leave the class before you understand all the steps in this lab.