

# Learning How to Code in Arduino

## Worksheet 3: Functions

Jason Krugman – Physical Computing - Fall 2012



### What is a function and what's it good for?

*A function is a group of code that we can call upon to do a given task. A function is something that we create and call on to perform... a function!*

We are already using functions in our Arduino coding. This worksheet will explain how to create customized functions that we can use to achieve a specific task.

*void loop* and *void setup* are both functions that we are already using. So are *pinMode*, *delay*, *digitalWrite*, and many others.

The “void” in front of “loop” and “setup” can be a little confusing. It has to do with the fact these functions do not need to “return” anything. Most functions work in the given manner: you give them some input variables, called “arguments”, they do calculations with those variables, and then they “return” the calculations to you. Both loop and setup are functions that are built into the way Arduino operates. They are actually defined within the Arduino architecture and must be “called” for every Arduino program to run. For more info, see this paper:

[http://web.cecs.pdx.edu/~EAS199/A/notes/06/whats\\_this\\_void\\_loop.pdf](http://web.cecs.pdx.edu/~EAS199/A/notes/06/whats_this_void_loop.pdf)

For our purposes, understanding the use of “void” is not terribly important. We just have to follow these simple guidelines: “void” always goes before setup and loop, and when we are defining one of our own functions. We do not have to write “void” before a function’s name when we are “calling” it during loop. “Calling” means calling it into action. We will get there shortly.

As we know, *void setup* runs once when we first start the Arduino, or press the reset button. Then *void loop* will continue to run continuously after that. Lets say we wanted to do something else when an event happens. For example, blink an LED once when a button is pressed. We can start off with something very simple, and gradually build up the complexity of the code using different programming methods. As the tasks you want the Arduino to do for you get more and more complex, these slightly more complicated programming methods will make things a whole lot easier in the long run.

Here is some basic code that does that makes an LED blink when a button is pressed using an IF statement:

*Sketch Name: led\_blink\_simple*

```
int buttonPin = 5; //button plugged to digital pin 5
int buttonVal = 0;
int ledPin = 3; //led plugged to digital pin 3
int del = 1000; //length of delay in milliseconds

void setup(){
  pinMode(buttonPin, INPUT);
  pinMode(ledPin, OUTPUT);
}

void loop(){
  buttonVal=digitalRead(buttonPin);
  if(buttonVal==HIGH){ //LED blinks twice when button is pressed
    digitalWrite(ledPin, HIGH);
    delay(del);
    digitalWrite(ledPin, LOW);
    delay(del);
    digitalWrite(ledPin, HIGH);
    delay(del);
    digitalWrite(ledPin, LOW);
    delay(del);
  }
}
```

**Now we will use a FOR loop to blink the LED 5 times every time the button is pressed:**

*Sketch Name: led\_blink\_with\_for*

```
int buttonPin = 5; //button plugged to digital pin 5
int buttonVal = 0;
int ledPin = 3; //led plugged to digital pin 3
int del = 1000; //length of delay in milliseconds
int numberOfTimes = 5; //number of times the LED will blink.

void setup(){
  pinMode(buttonPin, INPUT);
  pinMode(ledPin, OUTPUT);
}

void loop(){
  buttonVal=digitalRead(buttonPin);
  if(buttonVal==HIGH){
    for(int i = 0; i < numberOfTimes; i++){ //using a global variable to set number of blinks.
      digitalWrite(ledPin, HIGH);
      delay(del);
      digitalWrite(ledPin, LOW);
      delay(del);
    }
  }
}
```

## Now, we can take things a step further by using a function to blink the LED:

Sketch Name: *led\_blink\_function*

```
int buttonPin = 5; //button plugged to digital pin 5
int buttonVal = 0;
int ledPin = 3; //led plugged to digital pin 3

void setup(){
  pinMode(buttonPin, INPUT);
  pinMode(ledPin, OUTPUT);
}

void loop(){
  buttonVal=digitalRead(buttonPin);
  if(buttonVal==HIGH){
    blinkLED(5, 1000); //the arguments I put into the function determine number of blinks and delay
  }
} //loop is closed here

/*here is our custom function to blink LEDs.
The arguments we defined are numBlinks for number of blinks, and del for the delay between blinks.
We could put in any number of arguments to give the function more complex behaviors.
*/
void blinkLED(int numBlinks, int del){
  for(int i=0; i<numBlinks; i++){
    digitalWrite(ledPin, HIGH);
    delay(del);
    digitalWrite(ledPin, LOW);
    delay(del);
  }
}
```

Notice how the name for our function, “blinkLED” shows up twice in the above program: once when we “call” the function and once where we define the function. It seems a little weird that we “call” the function before we actually define it. The loop reaches the line where we say, “blinkLED(5,1000);” and it goes looking for this function called “blinkLED”. If it finds it, it follows the instructions we have laid out in *void blinkLED*. Remember, we use the word “void” before the function’s name when we are first defining it. This is similar to the way we put “int” before a variable we are defining for the first time.

Besides defining and calling functions, how to use arguments and pass variables. Once we call the function “blinkLED”, it takes those two numbers we have “passed into it”, the arguments 5 and 1000, and applies them to the function. It says to itself, “Okay 5, you are going to be stored in a **local variable** ‘numBlinks’, and 1000, you are going to be stored in a **local variable** ‘del’ ”.

“Local variable” means that these variables will only be used within the blinkLED function. If we say “numBlinks” somewhere else, like in the loop, the Arduino will say “ERROR”. This is what the function will effectively be doing with those two variables we have passed in:

```
for(int i=0; i<5; i++){
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```

This is awesome! It means that we can “pass” in numbers to our functions and have the functions do lots of work for us. We could even use a counter variable or a FOR loop to keep passing in lots of different numbers! The function will do the same thing to whatever argument variables we pass in to it. We can put as many or as few arguments into our functions as we like. See below.

**Now, we will start to take advantage of our use of functions:**

*Sketch Name: led\_blink\_function\_with\_for*

```
int buttonPin = 5; //button plugged to digital pin 5
int buttonVal = 0;
int ledPin = 3; //led plugged to digital pin 3

void setup(){
  pinMode(buttonPin, INPUT);
  pinMode(ledPin, OUTPUT);
}

void loop(){
  buttonVal=digitalRead(buttonPin);
  if(buttonVal==HIGH){

    for(int i=0; i<501;i+=100){
      blinkLED(3,i);
    } //end for loop
    /*by incrementing i by 100 each time we go through the loop,
    the del variable passed in to the blinkLED function will go up
    by 100. How many times will the blinkLED function get called per
    cycle of the loop? How many blinks will there be each time the
    function is called?"
    */
  } //end if statement
} //loop is closed here

/*here is our custom function to blink LEDs.
The arguments we defined are numBlinks for number of blinks, and del for the delay between blinks.
We could put in any number of arguments to give the function more complex behaviors.
*/
void blinkLED(int numBlinks, int del){
  for(int i=0; i<numBlinks; i++){
    digitalWrite(ledPin, HIGH);
    delay(del);
    digitalWrite(ledPin, LOW);
    delay(del);
  }
}
```

**All code is available for download at:**

[http://jasonkrugman.com/classes/physcomp/worksheet3/led\\_blink\\_simple/led\\_blink\\_simple.ino](http://jasonkrugman.com/classes/physcomp/worksheet3/led_blink_simple/led_blink_simple.ino)

[http://jasonkrugman.com/classes/physcomp/worksheet3/led\\_blink\\_with\\_for/led\\_blink\\_with\\_for.ino](http://jasonkrugman.com/classes/physcomp/worksheet3/led_blink_with_for/led_blink_with_for.ino)

[http://jasonkrugman.com/classes/physcomp/worksheet3/led\\_blink\\_function/led\\_blink\\_function.ino](http://jasonkrugman.com/classes/physcomp/worksheet3/led_blink_function/led_blink_function.ino)

[http://jasonkrugman.com/classes/physcomp/worksheet3/led\\_blink\\_function\\_with\\_for/led\\_blink\\_function\\_with\\_for.ino](http://jasonkrugman.com/classes/physcomp/worksheet3/led_blink_function_with_for/led_blink_function_with_for.ino)