
Aberystwyth Robotics Club - Arduino Tutorials - Functions Tutorial

Introduction

No prior worksheets need to have been completed to attempt this tutorial. There is a different version of the Functions Tutorial that extends from the ARC Arduino Blink, Button, LDR and Ultrasonic standalone tutorials.

This tutorial will show you how to bring code together into a manageable and tidy format.

Functions

When you are building robots, you need to get two halves of the system right: the hardware and the program that tells the hardware what to do.

If you have code that does lots of things (like a robot), it is useful to break the code into what are called "Functions".

You have two functions in your code already. One is called `setup()`, this is used to describe the hardware setup (what component is attached to what pin, what is an input and what is an output). The other function is called `loop()` this repeats the code again and again until you pull out the power.

You can define your own functions, here is an example with blinking an LED:

```
const byte ledPin = 13;
void setup() {
  pinMode(ledPin, OUTPUT);
}
void loop() {
  //do things before the blinkLED
  blinkLED();
  //do things after the blinkLED
}
void blinkLED() { //enter function
  digitalWrite(ledPin, HIGH); //sets the LED on
  delay(1000); //waits for a second
  digitalWrite(ledPin, LOW); //sets the LED off
  delay(1000); //waits for a second
} //return to where it was called
```

The example in the code here does not take anything in, and does not give anything in return. It just blinks the LED.

Where functions are really useful is when they package up a bit of code that does a specific job: moves forward a bit, or reads a sensor.



Other Example Functions

You can write functions that take numbers and do things with them. For example this function adds two numbers:

```
int add(int x, int y) {  
    int answer = x + y;  
    return answer;  
}
```

To use this function you would put this in a loop

```
int ans = add (3, 4);
```

and after that, the variable ans contains 7.

You can use this kind of pattern to write functions and read sensors. The important bit is the return value, which tells the function what it is going to give back.