**Buzzer**

|  |
| --- |
| void **setup**(){ //Nothing to see here, move along }  int toneDuration = 3; //In milliseconds  int buzzerPin = 2  void **loop**()  {  for (int freq = 400; freq<1500; freq+=10) {  tone(buzzerPin, freq);  delay(toneDuration\*1.3);       noTone(buzzerPin);    }  } |

**DigitalRead**

|  |
| --- |
| int buttonPin = 2;  int ledPin = 13;  void **setup**()  {    pinMode(buttonPin, INPUT\_PULLUP);    pinMode(ledPin, OUTPUT);  }  void **loop**()  {    int buttonState = digitalRead(buttonPin);    digitalWrite(ledPin, buttonState);  } |

**DigitalRead – Toggle an LED**

|  |
| --- |
| int buttonPin = 2; // the number of the pushbutton pin  int ledPin = 13; // the number of the LED pin  int ledState = HIGH; // the current state of the output pin  int buttonState; // the current reading from the input pin  void **setup**() {  pinMode(buttonPin, INPUT\_PULLUP);  pinMode(ledPin, OUTPUT);  }  void **loop**() {  int reading = digitalRead(buttonPin);  // if the button state has changed:  if (reading != buttonState) {  buttonState = reading;  // only toggle the LED if the new button state is LOW  if (buttonState == LOW) {  ledState = !ledState;  }  }     digitalWrite(ledPin, ledState); // set the LED  } |

**AnalogRead**

|  |
| --- |
| void **setup**() { //These are not the droids you are looking for }  int potentiometerPin = A0;  void **loop**() {     int potentiometerVal = analogRead(potentiometerPin);     int outputVal = map(potentiometerVal, 0, 1024, 0, 255);     analogWrite(10, outputVal);  } |

**Debounce**

|  |
| --- |
| int buttonPin = 2; // the number of the pushbutton pin  int ledPin = 13; // the number of the LED pin  int ledState = HIGH; // the current state of the output pin  int buttonState; // the current reading from the input pin  int lastButtonState = LOW; // the previous reading from the input pin  long lastDebounceTime = 0; // the last time the output pin was toggled  long debounceDelay = 50; //the debounce time; increase if output flickers  void **setup**() {  pinMode(buttonPin, INPUT\_PULLUP);  pinMode(ledPin, OUTPUT);  }  void **loop**() {  int reading = digitalRead(buttonPin);  // If the switch changed, due to noise or pressing  if (reading != lastButtonState) {  lastDebounceTime = millis(); // reset the debouncing timer  }  if ((millis() - lastDebounceTime) > debounceDelay) {  // whatever the reading is at, it's been there for longer  // than the debounce delay, so take it as the actual state      // if the button state has changed      if (reading != buttonState) {  buttonState = reading;  // only toggle the LED if the new button state is LOW  if (buttonState == LOW)  ledState = !ledState;        }      }    }    digitalWrite(ledPin, ledState); // set the LED  lastButtonState = reading; // save the reading  } |

# Common functions

digitalRead(int pin)

This function is used to read the state of the specified pin. It will return either LOW or HIGH.

analogRead(int pin)

This function returns a value between 0 and 1024. This represents the voltage on the specified pin between 0v and 5v.

map(int value, int fromLow, int fromHigh, int toLow, int toHigh)

Maps one range of values to another and returns the result. Useful for mapping analogue inputs to PWM outputs.

e.g. map(value, 0, 1024, 0, 255);

millis()

Returns the number of milliseconds the Arduino has been running. This returns a value of type long.