Photoresistor - While Loop

Sometimes you want everything in the program to stop while a given condition is true. You can do this using a [while loop](https://www.arduino.cc/en/Reference/While). This example shows how to use a while loop to [calibrate](https://www.arduino.cc/en/Tutorial/Calibration) the value of an analog sensor.

In the main loop, the sketch below reads the value of a photoresistor on analog pin 0 and uses it to fade an LED on pin 9. But while a button attached to digital pin 2 is pressed, the program runs a method called calibrate() that looks for the highest and lowest values of the analog sensor. When you release the button, the sketch continues with the main loop.

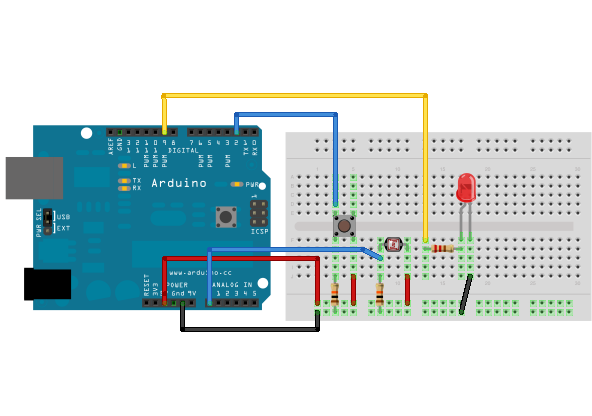
This technique lets you update the maximum and minimum values for the photoresistor when the lighting conditions change.

Hardware Required

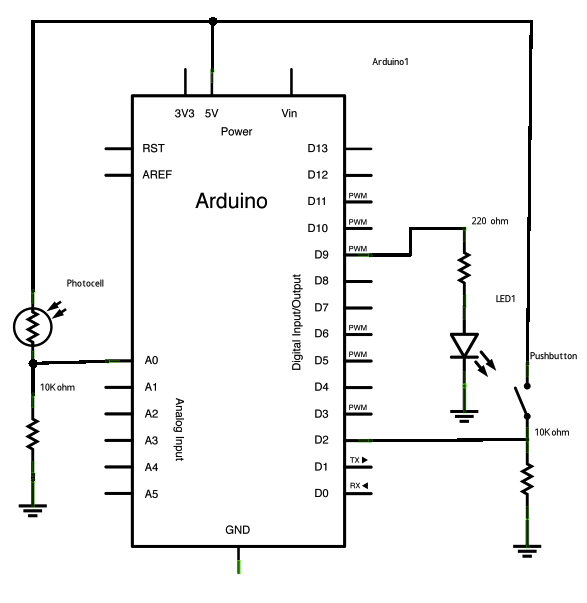
* Arduino or Genuino Board
* pushbutton or switch
* photoresistor or another analog sensor
* 2 10k ohm resistors
* breadboard

Circuit

Connect your analog sensor (e.g. potentiometer, light sensor) on analog input 2 with a 10K ohm resistor to ground. Connect your button to digital pin, again with a 10K ohm resistor to ground. Connect your LED to digital pin 9, with a 220 ohm resistor in series.



Schematic



Code

*// These constants won't change:*  
const int sensorPin = A0;       *// pin that the sensor is attached to*  
const int ledPin = 9;           *// pin that the LED is attached to*  
const int indicatorLedPin = 13; *// pin that the built-in LED is attached to*  
const int buttonPin = 2;        *// pin that the button is attached to*  
  
  
*// These variables will change:*  
int sensorMin = 1023;  *// minimum sensor value*  
int sensorMax = 0;     *// maximum sensor value*  
int sensorValue = 0;         *// the sensor value*  
  
  
void **setup**() {  
  *// set the LED pins as outputs and the switch pin as input:*  
  pinMode(indicatorLedPin, OUTPUT);  
  pinMode(ledPin, OUTPUT);  
  pinMode(buttonPin, INPUT);  
}  
  
void **loop**() {  
  *// while the button is pressed, take calibration readings:*  
  while (digitalRead(buttonPin) == HIGH)

{  
    calibrate();  
  }  
  *// signal the end of the calibration period*  
  digitalWrite(indicatorLedPin, LOW);  
  
  *// read the sensor:*  
  sensorValue = analogRead(sensorPin);  
  
  *// apply the calibration to the sensor reading*  
  sensorValue = map(sensorValue, sensorMin, sensorMax, 0, 255);  
  
  *// in case the sensor value is outside the range seen during calibration*  
  sensorValue = constrain(sensorValue, 0, 255);  
  
  *// fade the LED using the calibrated value:*  
  analogWrite(ledPin, sensorValue);  
}

void calibrate() {  
  *// turn on the indicator LED to indicate that calibration is happening:*  
  digitalWrite(indicatorLedPin, HIGH);  
  *// read the sensor:*  
  sensorValue = analogRead(sensorPin);  
  
  *// record the maximum sensor value*  
  if (sensorValue > sensorMax) {  
    sensorMax = sensorValue;  
  }  
  
  *// record the minimum sensor value*  
  if (sensorValue < sensorMin) {  
    sensorMin = sensorValue;  
  }  
}