Arduino Basics

An Arduino tutorial blog. Free Arduino tutorials for everyone !

19 August 2011

Poor Man's Colour Detector (Part 2) - The project

In this project we will be detecting the colour of 3 different Mega Blok colours (Red, Yellow and Green). We will be using an Arduino UNO connected to 2 LEDs (one Yellow and one Red LED) as light detectors, and an RGB LED to illuminate the subject. We will use a Photocell to account for varying ambient light levels.

The signals from the LED light sensors will be sent to a Processing.org program via a Serial command. The computer program will make use of my Neural Network to classify the pattern of results and hopefully provide the correct colour "answer". The program should change the colour of the computer screen background to coincide with the colour of the Mega Blok.

The Video

Parts Required:

Arduino UNO.....x1

Red LED x1

Yellow LED.....x1

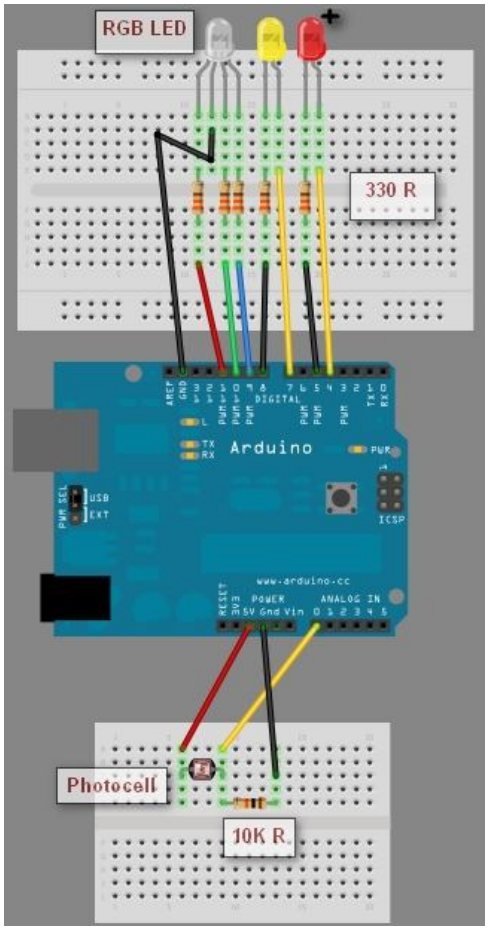
330 Ohm resistors... x 5 (for the LEDs)

Photocell x1

10K Ohm resistor....x1 (for the Photocell)

Around 11 wires and a Breadboard (or two) to put it all together

Here is the Fritzing Sketch: (made with Fritzing)



The Arduino Code

Load the following code into the Arduino.

arduino code Arduino: Colour Detector

```
/* Define the pin for the PhotoResistor */
#define PhotoR_Pin 0

/* Define the pins for the Red LED Sensor */
#define Red_LED_Sensor_POS 4
#define Red_LED_Sensor_NEG 5

/* Define the pins for the Yellow LED Sensor */
#define Yellow_LED_Sensor_POS 7
#define Yellow_LED_Sensor_NEG 8

/* Define the pin for the RGB LED torch */
#define RGB_LED_RedPin 11
#define RGB_LED_GreenPin 10
#define RGB_LED_BluePin 9

/* Controls the brightness of the RGB LED */
int intensity=255;


/* Define the maximum cycles/time allowed for each LED to capture light */
long max_darkness=80000;


void setup(){
  /* Setup the RED LED Sensor */
  pinMode(Red_LED_Sensor_POS,OUTPUT);
  digitalWrite(Red_LED_Sensor_POS,LOW);

  /* Setup the YELLOW LED Sensor */
  pinMode(Yellow_LED_Sensor_POS,OUTPUT);
  digitalWrite(Yellow_LED_Sensor_POS,LOW);

  /* No need to setup the RGB LED Pins */
```

```
36
37    /* Turn on Serial Protocol */
38    Serial.begin(9600);
39  }
40
41  void loop()
42  {
43    byte byteRead;
44
45     /*  check if data has been sent from the computer: */
46    if (Serial.available()) {
47
48      /* read the most recent byte (which will be from 0 to 255): */
49      byteRead = Serial.read();
50
51      if(byteRead==0){
52
53        /* Turn off if the byte Read was 0 */
54        set_RGB_LED(0,0,0,false);
55
56      }else{
57
58        /* set the brightness of the LED and then take readings: */
59        set_RGB_LED(0,0,0,false);
60        photoR_Read();
61        set_RGB_LED(0,0,0,true);
62        set_RGB_LED(intensity,0,0,true);
63        set_RGB_LED(0,intensity,0,true);
64        set_RGB_LED(0,0,intensity,true);
65      }
66    }
67  }
68
69  void photoR_Read(){
70    int ambiLight = analogRead(PhotoR_Pin);
71    ambiLight = map(ambiLight, 0, 900, 0, 50);
72    ambiLight = constrain(ambiLight, 0, 50);
73
74     /* Print the Ambient light level to the serial port */
75    Serial.println(ambiLight);
76  }
77
78  void set_RGB_LED(int redInt, int greenInt, int blueInt, boolean takeReadings ){
79    /* set the brightness and colour of the RGB LED: */
80      analogWrite(RGB_LED_RedPin, redInt);
81      analogWrite(RGB_LED_GreenPin, greenInt);
82      analogWrite(RGB_LED_BluePin, blueInt);
83
84    /* If takeReadings is true - then take Readings. */
85    if(takeReadings){
86
87      /* Read the amount of Yellow light */
88      read_LED('Y', Yellow_LED_Sensor_NEG);
89
90      /* Read the amount of Red light */
91      read_LED('R', Red_LED_Sensor_NEG);
92    }
93  }
94
95  void read_LED(char LED_Colour, int LED_Pin){
96
97    /* Charge the LED by applying voltage in the opposite direction */
98    pinMode(LED_Pin,OUTPUT);
99    digitalWrite(LED_Pin,HIGH);
100
101   /* Read the amount of Light coming into the LED sensor */
102   long darkness=0;
103   int lightLevel=0;
104   pinMode(LED_Pin,INPUT);
105   digitalWrite(LED_Pin,LOW);
106
107   while((digitalRead(LED_Pin)!=0) && darkness < max_darkness){
108      darkness++;
109   }
110
111   lightLevel=((max_darkness-darkness)+1)/80;
112
113   /* Print the light level to the serial port */
114   Serial.println(lightLevel);
  }
```

code formatter

The Processing Code:

The processing code is very long:
Please visit this link to copy and paste the code into your Processing sketch.
http://www.openprocessing.org/visuals/?visualID=34210

Make sure to select "Source Code" when you get there: (as displayed below)

If you have any problems with accessing the code - please let me know in the comments section of this blog.

This sketch utilises a simple feed forward Neural Network (that I developed from scratch). For more detailed information about this neural network please navigate through my previous blog postings.

Neural Network

Part 1 : The Connection class

Part 2 : The Neuron class

Part 3 : The Layer class

Part 4 : The Neural Network class

Part 5: Back Propagation - the process

Part 6 : Back Propagation (A complete walk through)

So there you go, a simple idea, a simple outcome, and a lot of "stuff" happening in the background.
I am sorry. This project is not basic, but hopefully someone out there will get some use out of it.

Have fun !

ScottC

Posted by Scott C at 22:11 G+1 Recommend this on Google

Labels: Arduino, ArduinoBasics, best arduino blog, code, color, Colour, LED as Sensor, Mega Bloks, Neural Network, PhotoCell, Processing, project, UNO

10 comments:

Anonymous 12 November 2012 at 16:47

Great project, thank you !!!!
I have read your ANN for processing and this project, but I have a question, you have this in your code >

lightLevel=((max_darkness-darkness)+1)/80;

and send it to PC using USB, in your neural network depending on the top buttons in your sketch will be the inputs right, but regarding the measured value, where did it go ? which part in you procesing sketch of your neural network ? I want to know how you use the measeured values in your neural network, you multiply it by what ? It is a little confusing, the value of the top buttons are inputs, but the inputs of the measured values where did it go ? Can you please answer my question, thanks

Reply

Replies

ScottC 12 November 2012 at 20:55

Hi Anonymous,
It has been a very long time since I have written this code... so had to refresh my memory on how this was pieced together. And looking through it, I can see how easy it is to get entangled in the code. Knowing what I know now, I probably would have approached it a little differently, but I will try my best to explain what is happening.

Step 1: You press the top buttons, which sends a signal to the Arduino to shine the RGB LED, and it in turn will send back 3 values. One for the ambient light taken by the photoresistor:

Serial.println(ambiLight);

One for the Yellow LED, and one for the Red LED.
Serial.println(lightLevel);

Ok - Now processing will be listening for the return of 3 values, which it will place into an array called sensorReadings.
This is achieved by the "void serialEvent (Serial myPort)" function.
Once it has all 3 values, it will then process these values in one of 2 ways. If the "detecting" boolean value is true, it will use the neural network to try and detect the colour. If it is false, then it will send these readings to a file "outputFile1", as well as sending the expected results to "outputFile2". The expected results are pre-defined and based on the button pressed.

So first you need to go through an show it what ambient looks like, what a yellow block looks like, what a green block looks like etc etc by pressing the top buttons. Once these readings are saved in a file. You press the train network button. This loads the values from outputFile1 and 2, into the neural network. Once trained, you can then test it out by trying to detect.
Bear in mind, you may need to experiment under varying light conditions. For example, if you train the network at night time, it may have trouble during the day, but it is pretty quick to retrain.

Reply

Luis Enrique Mendez Hidalgo 17 November 2012 at 11:52

I need the executable where cruise buttons red, green, yellow etc. or I can explain how I do it?

Reply

Replies

ScottC 18 November 2012 at 14:52

Hi Luis,
Am not too sure I understand what you are asking. Can you please try again?

Reply

Ash 3 February 2014 at 09:28

Sir,

Good Day I just want to ask how to make this work..

This is how I do it

1st I create the circuit

2nd I input the codes into my arduino..

The question is how I could add the Processing Code..?

Please help.. I'm a beginner in using arduino. Thank you for your kind consideration.

Reply

Replies

Scott C 3 February 2014 at 15:03

Hi Ash - you chose an interesting project for a beginner.
Processing is a program that you run on your computer. The processing program communicates to the Arduino via USB (serial COM port).
Basically, the Arduino and LEDs are the eyes of this project, and your computer is used as the brain (neural network).
So step 3 is download the processing IDE from here
And copy and paste the code from the link here - making sure to select the code tab as indicated in the instructions above, and paste the code into the processing IDE.

Make sure that your Arduino is connected to the computer via USB.
Ensure that you close any Serial Monitors that may be open. Do this before you run the processing code.

Step 4 - train the neural network as shown in the video - you need to place a red coloured megablok over the LEDs when training with the red button, etc etc.

Step 5:
Once trained - you can then see how effective your neural network is by trying to detect each of the colours.

Tip: try an keep the lighting conditions consistent... for example, if you train it in bright sunlight, don't expect it to detect your colours in your room at night. Ambient light is the biggest problem you will face in terms of consistent results.
But I guess if you train the network under different conditions, you may bet better results. It requires a bit of trial and error,

This project was one of my first, and was done out of curiosity, which probably explains the lack of detail.

Regards
Scott

Reply

Anonymous 4 February 2015 at 18:06

hi Scott, this question my rack your brain a bit, but here goes. Is it possible to adapt the Processing code so it can run on an Arduino ? I would like to keep the NN on my mobile platform, without the need for a PC connection.

regards, Gert

Reply

Replies

Scott C 5 February 2015 at 08:50

Hi Gert,

It would be easier to start from scratch than it would be to adapt the code. However, it is possible to develop a NN on an Arduino without Processing. I used the processing language because it allowed me to take advantage of my computer's processing power and

memory. It also allowed me to create a GUI without much effort.
Hopefully there is enough explanation within these tutorials to get you on the right track, however there are many others who have done what you are trying to do... a google search is likely to get you there pretty quickly.
Good luck

Reply

Merquiades Betanco 5 February 2015 at 00:28

Hello project is super great but I have a problem and I need your help please look at that when I run the code prossesig the window that appears to me is a tiny gray buttons bottomless help me it is urgent for my project dime electronics to modify or do it then appears my window with buttons

Reply

Replies

Scott C 5 February 2015 at 08:54

Hi Merquiades,
I made this tutorial in 2011. It is very likely that the Processing version was different back then, and may not work with the current version. The processing website has many tutorials on how to create buttons etc etc, and once you have the GUI set up, the rest should be easy to adapt.

Reply

Enter your comment...

Comment as: Unknown (Goo ▾ Sign out

Publish Preview Notify me

Feel free to leave a comment about this tutorial below.
Any questions about your particular project should be asked in the ArduinoBasics forum.

Comments are moderated due to large amount of spam.

Links to this post

Create a Link

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)

This Week's Most Popular Posts Most Recent Posts

433 MHz RF module with Arduino Tutorial 1

HC-SR04 Ultrasonic Sensor

Simple Arduino Serial Communication

Relay Module

433 MHz RF module with Arduino Tutorial 2

Get Arduino Data over the internet using jQuery and AJAX

Two Million Views

Generosity Campaign Update - Day 3

Generosity Campaign Update - Day 2

Generosity campaign - Day 1

Recent Posts Widget by Helplogger