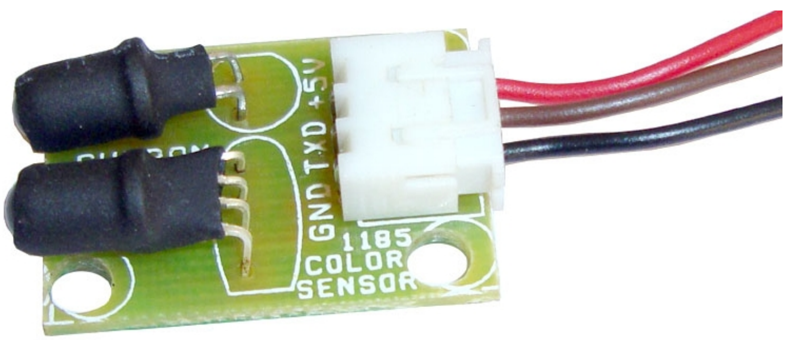
Color Sensor

This color sensor identifies color and gives serial output of RBG value. It can identify 16.7 million color shades giving RGB value for the detected color. The detected color is identified as amount of three primary color values namely Red, Green & Blue with 8 bit accuracy for each primary color. Any color can be separated or combined into three primary colors Red, Green and Blue using the RBG values.



Features

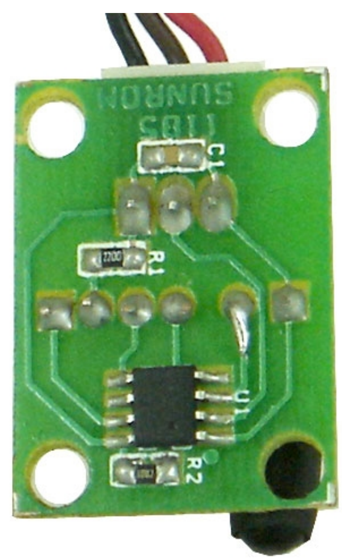
- Individual RGB color detected
- Simple 5V operation
- Serial data output for complete RGB values
- UART interface for direct connection to any MCU or USB-TTL convertor

Applications

- Color Detection & Sorting operations
- Process control to printed materials
- Ambience light detection / Robotics color detection

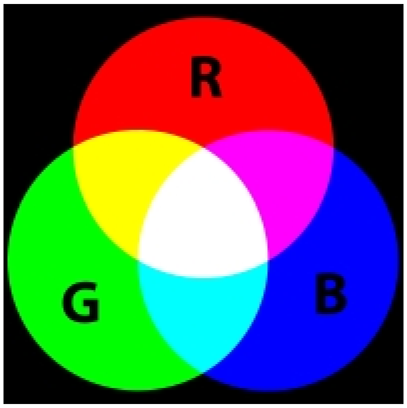
Specification

Parameter	Value	Unit	Notes
Operating Voltage	5	V DC	Provide regulated 5V supply
Current	20	mA	
Color Detecting Capacity	16.7 millions	RGB	R=8 bit (2^8=256 levels) G=8 bit (2^8=256 levels) B=8 bit (2^8=256 levels) 256x256x256=16.7 millions shades detection
Color measuring range	350-750	Nm	
Luminance range	100	Lux	
Response time	500	ms	
Output Data baud	9600	Bps	5V level output UART Properties (8-N-1) Start bit:1 bit Data bit: 8 bits Parity: None Stop bit: 1 bit



Principle of Color Identification

The sensor switches each primary color RGB, one by one and checks what intensity of color is reflected by the surface of detection. This reflected intensity is converted to 8 bit value. For example a RED surface will strongly reflect RED. While a Yellow surface will reflect RED and GREEN both. According to the induction principle of the three primary colors which create various other colors in nature, once the value of three primary colors is confirmed, the color of the tested object is known. Knowing the value of RGB helps people gain the color of the light which is projected onto the sensor since each color correspond to only one value of RGB.



Further details on RGB model is here
http://en.wikipedia.org/wiki/RGB_color_model

Serial Data Output format

The serial data at 9600 baud rate consist of 25 bytes for each 500ms interval.

When RED shade of color is detected you would get following type of data in terminal
R=130 G=030 B=030 L=010

Here value of RED is 130 while Green and Blue are 30 both
L=10 means the amount of Light reflected by surface, White surface will reflect most and black the least, This L value you can use to detect the darkness of surface. We recently added this L parameter since it was difficult to detect white and black surface from only RGB values. The sample code of microcontroller and VB software does not implement L value processing but it works with only RGB values. L value can be used to detect white/black surface.

Each value will be from 0 to 255, Let us see each byte in detail

Count	HEX Value	ASCII	Notes
1	0x0D	\r	Carriage return character. Can also use as Start of packet identifier
2	0x52	R	Always 'R' character
3	0x3D	=	Always '=' character
4	0x31	1	Red Value Hundreds Character ASCII, Will be between 0-9
5	0x33	3	Red Value Tens Character ASCII, Will be between 0-9
6	0x30	0	Red Value Ones Character ASCII, Will be between 0-9
7	0x20		Always Space Character
8	0x47	G	Always 'G' character
9	0x3D	=	Always '=' character
10	0x30	0	Green Value Hundreds Character ASCII, Will be between 0-9
11	0x33	3	Green Value Tens Character ASCII, Will be between 0-9
12	0x30	0	Green Value Ones Character ASCII, Will be between 0-9
13	0x20		Always Space Character
14	0x42	B	Always 'B' character
15	0x3D	=	Always '=' character
16	0x30	0	Blue Value Hundreds Character ASCII, Will be between 0-9
17	0x33	3	Blue Value Tens Character ASCII, Will be between 0-9
18	0x30	0	Blue Value Ones Character ASCII, Will be between 0-9

13	0x20		Always Space Character
14	0x42	L	Always 'L' character
15	0x3D	=	Always '=' character
16	0x30	0	Light Value Hundreds Character ASCII, Will be between 0-9
17	0x33	1	Light Value Tens Character ASCII, Will be between 0-9
18	0x30	0	Light Value Ones Character ASCII, Will be between 0-9
19	0x0A	\n	New Line character. Can also use as End of packet identifier

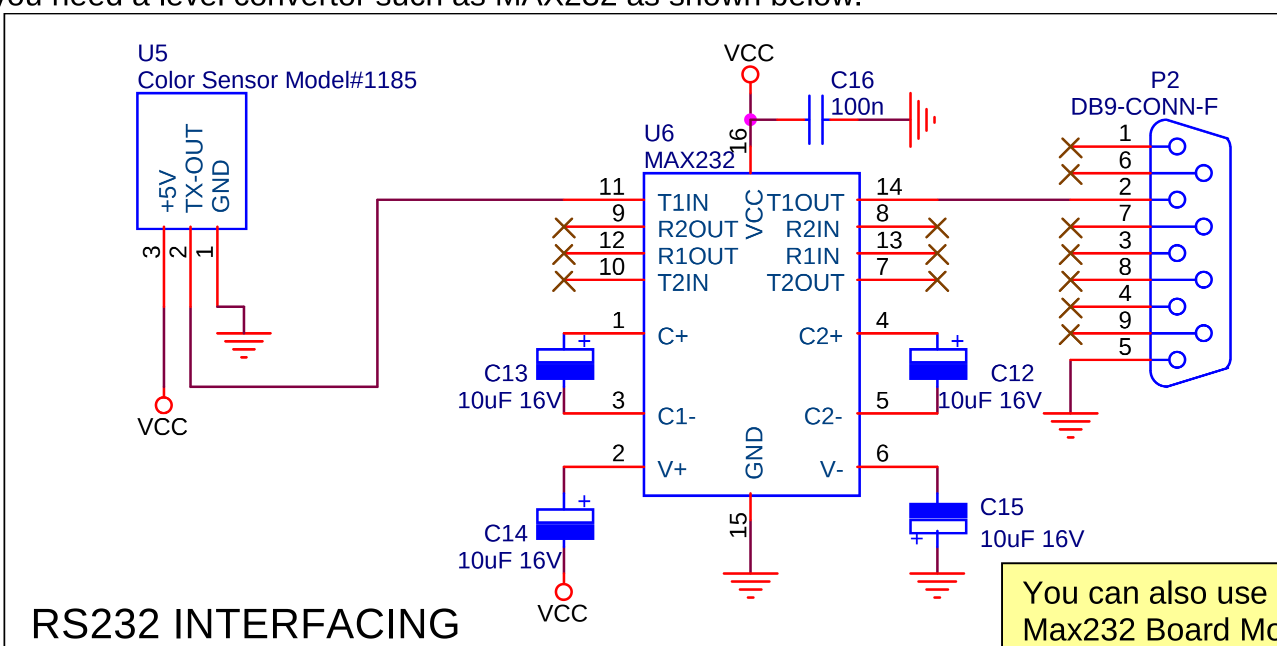
In examples below, we have shown how to parse this incoming data into integers using Microcontroller or PC software in .NET. Full source code is given for it.

Calibrating the sensor

The output you get for a red surface would contain R value the most out of RBG. It does not reflect the actual red value of surface. If you multiply the R value with a constant(scaling factor) then match with actual R value then you can get actual RGB values. This can done easily with software provided in VB. Once you calculate the actual RBG values by matching the color in VB with the surface color of material. You can use this multiplier value to scale the output to actual RBG values of material.

Interfacing with RS232

If you wish to interface the module with RS232 level like a PC serial port or any other RS232 device you need a level convertor such as MAX232 as shown below.

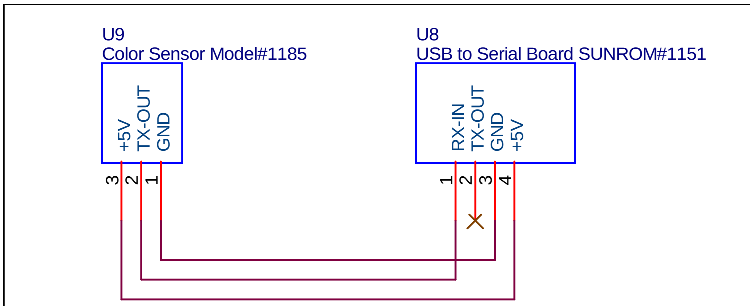


You can also use our Max232 Board Model 1104

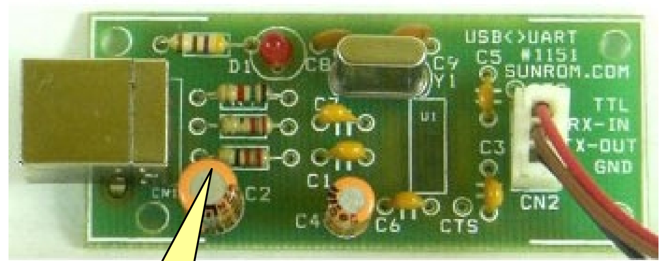
http://www.sunrom.com/p-245.html

Interfacing to USB Port and Powering from USB Port

You can use any USB to TTL convertor to convert the UART data of sensor through USB interface to PC.

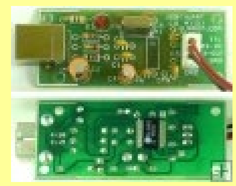


USB INTERFACING



It will appear as virtual serial port on PC to which you can communicate through any software which can transmit receive by this serial port like hyperterminal or custom made software.

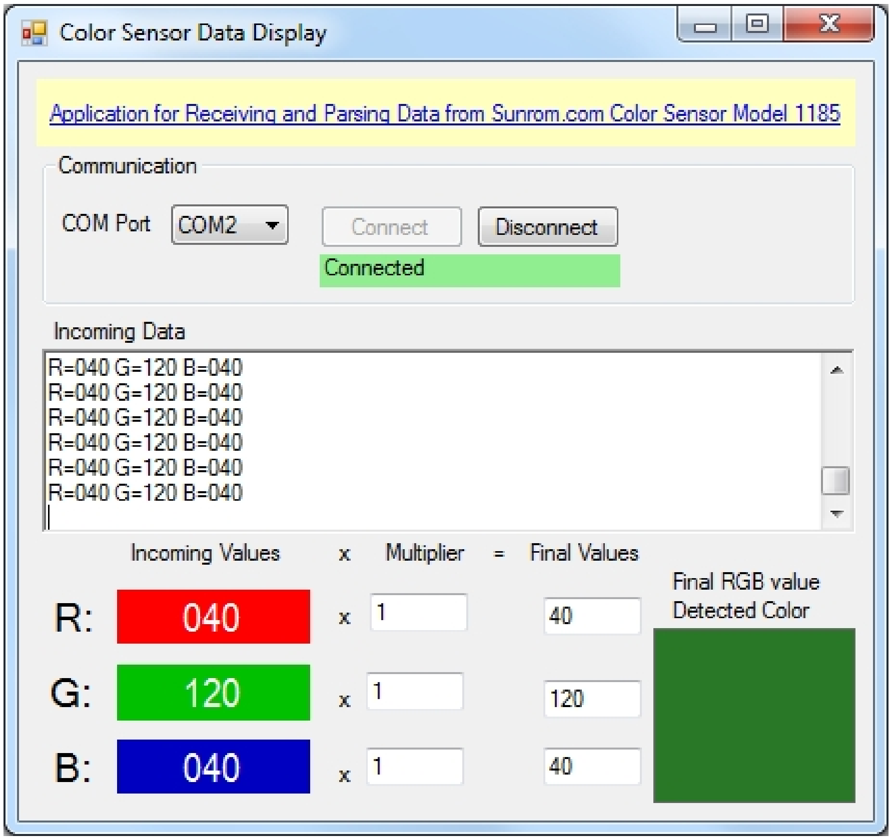
You can use our USB to Serial Board Model 1151



http://www.sunrom.com/p-244.html

To get +5V power for Senor from USB port, solder +5V wire of Sensor to +ve pin of this capacitor.

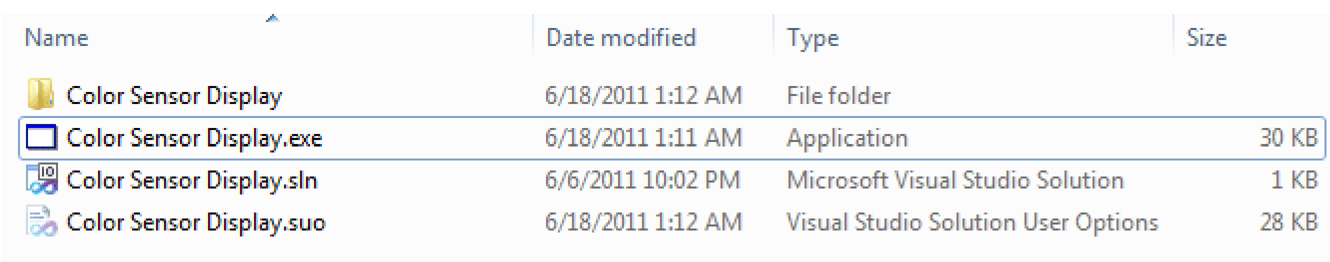
PC Software with Source Code



We created this software in VB.NET 2010 for reading the incoming data and parse the data into RGB values. Full source code is available for download for further changes. The RGB values are applied to a square box, this showing detected color. There is an ability to multiply detected color value to create a more distinct color shade. This feature can be used to adjust the detected color and with actual color shade.

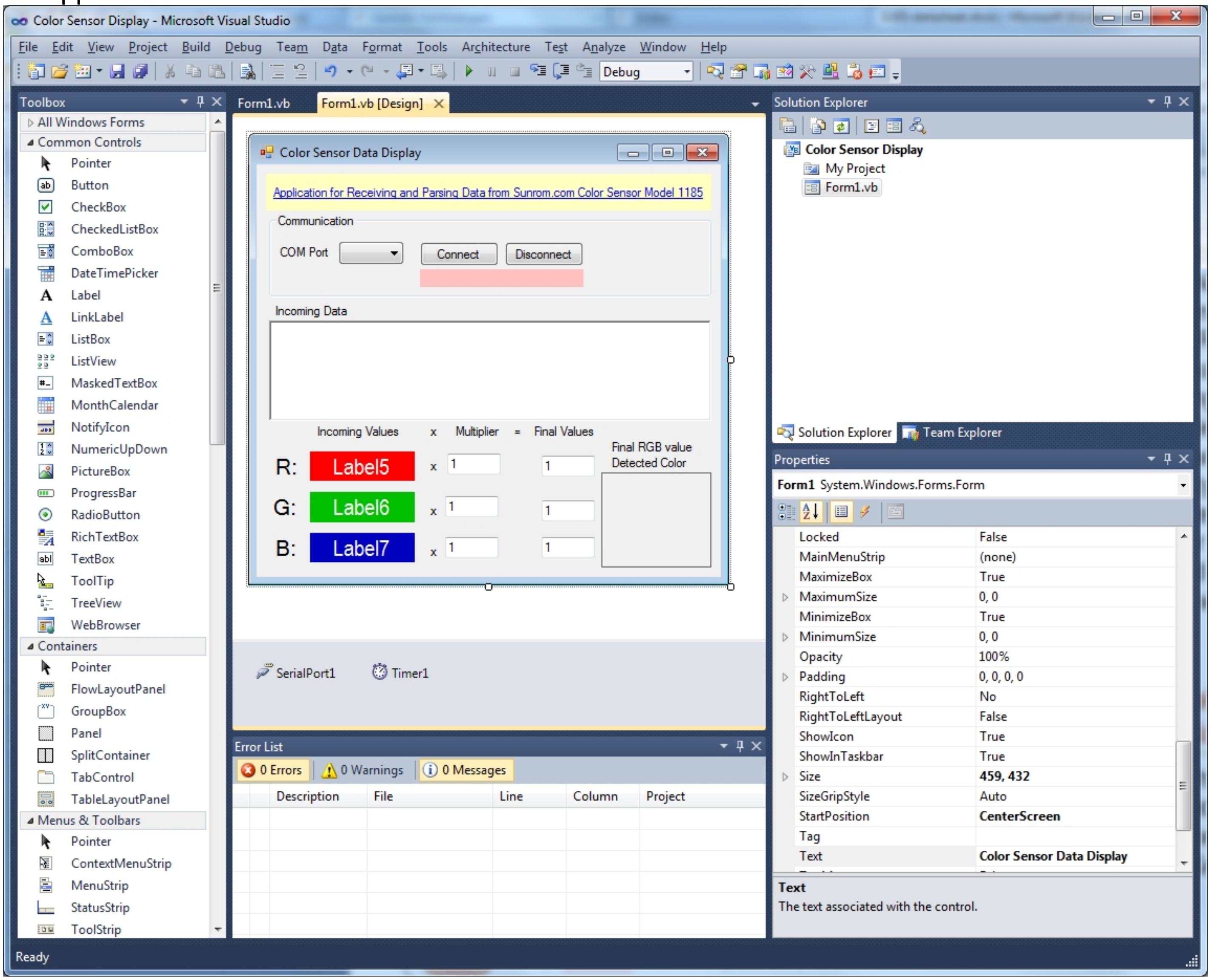
Download Application and its Source Code from this link
http://www.sunrom.com/files/1185-app.zip

After source code is unzip, you can get already compiled EXE file and Visual Studio project file which you can open in MS Visual Studio 2010. The folder has the source files for this project as shown below.



Name	Date modified	Type	Size
Color Sensor Display	6/18/2011 1:12 AM	File folder	
Color Sensor Display.exe	6/18/2011 1:11 AM	Application	30 KB
Color Sensor Display.sln	6/6/2011 10:02 PM	Microsoft Visual Studio Solution	1 KB
Color Sensor Display.suo	6/18/2011 1:12 AM	Visual Studio Solution User Options	28 KB

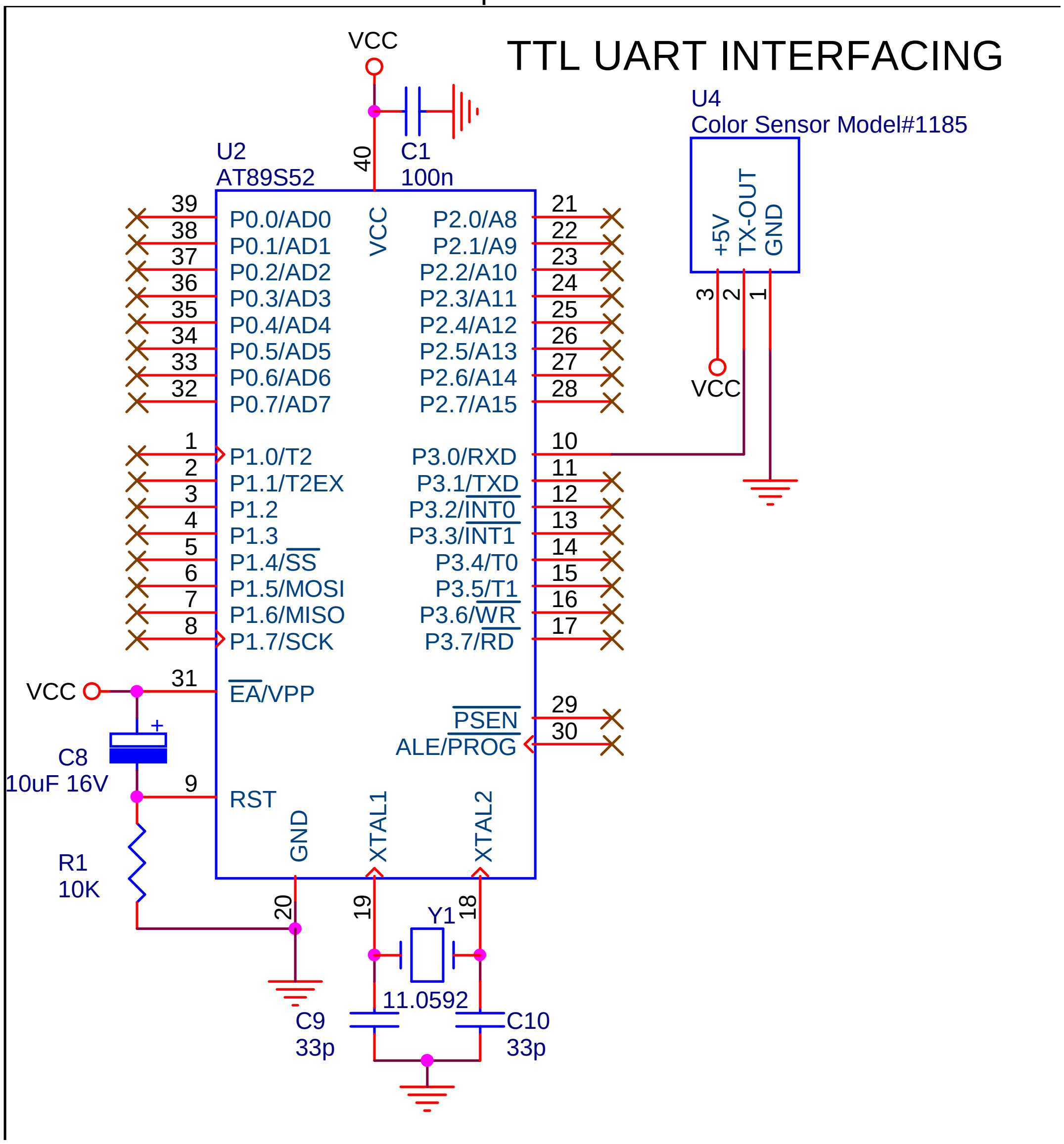
After project is open in Visual Studio you can modify whatever parameters you wish and develop the application further.



Read Color sensor data to MCU

We have used AT89S52's RXD pin to receive serial data from sensor. You can use any microcontroller to interface using this interface. We have chosen AT89S52 to show here since it is more widely used but can be any 8051 MCU or other MCU. The sample code we have given can be adapted to any C compiler or any microcontrollers like AVR or PIC since with minor changes.

The interfacing with microcontroller is shows below, The sensor level is 5V data so directly connected to Microcontroller RXD pin.



Sample Code is compiled using keil compiler given on next page.

```c
// Sample code to receive serial data from color sensor model 1185 from sunrom.com and
// separate into integer values.      Compiler: Keil
#include <REGX51.H> // standard 8051 defines
// -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
// -=-=-=-=- Hardware Defines -=-=-=-=-=-=-=-
// -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
char sbuffer[25], ch; // Array Holds incoming serial data
unsigned char pos;
unsigned char iR, iG, iB;
//receive serial character from serial port
char mygetchar(void)
{
      char c;
      while(!RI);
      RI =0;
      c = SBUF;
      return SBUF;
}
// -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
// -=-=-=-=- Main Program -=-=-=-=-=-=-=-
// -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
void main()
{
// -=-=- Intialize variables -=-=-=
      pos = 0;
// -=-=- Intialise Serial Port -=-=-=
      //Sets up MCU to use 9600 bps @ 11.059 MHz Crystal
      SCON = 0x52; // 8-bit UART mode
      TMOD = 0x20; // timer 1 mode 2 auto reload
      TH1= 0xfd; // 9600 8-n-1
      TR1 = 1; // run timer1
// -=-=- Program Loop -=-=-=
      while(1)
      {
            ch = mygetchar(); //loop till character received
            if(ch==0x0A) // if received character is <LF> end of line, time to display
            {
                  pos = 0; // buffer position reset for next reading
                  // extract data from serial buffer to 8 bit value
                  // convert data from ASCII to decimal:
                  // For example ASCII ‘1’ has HEX value of 0x31, to convert it to integer 1
                  // we have to minus 0x30 so 0x31-0x30 = 1  Here 0x30 is value of ASCII ‘0’
                  //        Hundred Digit          Ten Digit               One Digit
                  iR = ((sbuffer[3]-'0')*100) + ((sbuffer[4]-'0')*10) + (sbuffer[5]-'0');
                  iG = ((sbuffer[9]-'0')*100) + ((sbuffer[10]-'0')*10) + (sbuffer[11]-'0');
                  iB = ((sbuffer[15]-'0')*100) + ((sbuffer[16]-'0')*10) + (sbuffer[17]-'0');
                  // Do whatever you wish to do with these three integer variables
                  // Show on LCD or Do some action as per your application
                  // Value of iR, iG, iB will be between 0-255
                  // You can do something like below to switch on relay when red is detected
                        //if(IR > 100 && iG < 50 && iB < 50)
                              // RELAY = 1
            } else {
                  sbuffer[pos] = ch; //store serial data to buffer
                  pos++;
            }
      } // end while
}// end main
```

Sensor Dimensions