Color Detection Research Project

By

Neel Kabirpanthi
# Index

1. Introduction .......................................................................................................................... 3  

2. Description ............................................................................................................................ 4  
   a) The RGB Color Room.................................................................................................... 4  

3. Different Measures ............................................................................................................... 6  
   a) Measures with different distance to the object............................................................. 6  
   b) Measures with different angle of the sensor................................................................. 7  
   c) Temperatures measurement.......................................................................................... 8  
   d) Temperatures measurement of the monitor diode......................................................... 9  

4. Researches of methods for color identification .................................................................. 13  
   a) First method.................................................................................................................. 14  
   b) Second method............................................................................................................. 15  
   c) Third method................................................................................................................ 15  
   d) Comparison of the three methods................................................................................. 15  
   e) Last method for color identification............................................................................. 16  

5. Selection of the construction units - Hardware ............................................................... 17  
   a) Introduction.................................................................................................................. 17  
   b) Power supply............................................................................................................. 17  
   c) Sensor ........................................................................................................................ 17  
   d) The Amplifier ............................................................................................................. 18  
   e) Microcontroller ......................................................................................................... 18  
   f) Speech chip ............................................................................................................... 19  
   g) Schematic .................................................................................................................. 20  
   h) Board ......................................................................................................................... 20  

6. The design ........................................................................................................................... 22  

7. Introduction of Software Development ............................................................................ 23  
   a) The Program.............................................................................................................. 24
1. Introduction

The task of the project was to develop a device which can detect different colors by using the sensor... of the company. The target group of the device are blind persons or persons who can not detect different colors. There are already different systems on the market, but they are very expensive and not useful for affordable.
The new device should be a compact, simple and a cheap solution.
2. Description

The sensor consists of three LED (red, green, blue) two Pin Diodes and a Monitor Diode. Consecutively on Led will be turned on and then the Pin Diodes are measuring the reflection of the light. Each color has a different reflection for the three LED. When the reflection is high the Pin Diodes have a small resistor, is there a little reflection the resistor is high. This value of resistance we transform into a voltage. So we get for each LED a value of voltage which will be converting into the RGB Values. This value were compared with the reference color (a collection of selected colors), and so the device get the right color. The name of the color is given back by a speaker.

a) The RGB Color Room

To get an idea how it works to detect the color, we give a short explain of the RGB Color Room.

The RGB Color Room is an additive Model. This means that the three fundamental colors (Red, Green, and Blue) can be added to white. Nearly each color can be described with these three fundamental colors.

In the graph we see the RGB Color Room. On the x-Axes is the color Red, on the y-Axes is the color Blue and on the z-Axes is the color Green. On the diagonal of the color room we find the grey scale. In the graph we can see that each color is a combination of the three fundamental colors.

The measurements were made, to get an idea, which are the important factors to detect the right color. For the measurements we used a System which contains a Lab VIEW-Program, DAC-Card and the sensor. As Colors we used the „Digital ColorChecker SG“of the company Gretagmacbeth, which consists of the standard colors and much more colors.

In the following table are the colors of the color Checker which we used for our measurements.
<table>
<thead>
<tr>
<th>Index</th>
<th>Koordinaten</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2-E</td>
</tr>
<tr>
<td>2</td>
<td>2-F</td>
</tr>
<tr>
<td>3</td>
<td>2-G</td>
</tr>
<tr>
<td>4</td>
<td>2-H</td>
</tr>
<tr>
<td>5</td>
<td>2-I</td>
</tr>
<tr>
<td>6</td>
<td>2-J</td>
</tr>
<tr>
<td>7</td>
<td>3-E</td>
</tr>
<tr>
<td>8</td>
<td>3-F</td>
</tr>
<tr>
<td>9</td>
<td>3-G</td>
</tr>
<tr>
<td>10</td>
<td>3-H</td>
</tr>
<tr>
<td>11</td>
<td>3-I</td>
</tr>
<tr>
<td>12</td>
<td>3-J</td>
</tr>
<tr>
<td>13</td>
<td>4-E</td>
</tr>
<tr>
<td>14</td>
<td>4-F</td>
</tr>
<tr>
<td>15</td>
<td>4-G</td>
</tr>
<tr>
<td>16</td>
<td>4-H</td>
</tr>
<tr>
<td>17</td>
<td>4-I</td>
</tr>
<tr>
<td>18</td>
<td>4-J</td>
</tr>
<tr>
<td>19</td>
<td>5-E</td>
</tr>
<tr>
<td>20</td>
<td>5-F</td>
</tr>
<tr>
<td>21</td>
<td>5-G</td>
</tr>
<tr>
<td>22</td>
<td>5-H</td>
</tr>
<tr>
<td>23</td>
<td>5-I</td>
</tr>
<tr>
<td>24</td>
<td>5-J</td>
</tr>
</tbody>
</table>
3. Different Measures

a) *Measures with different distance to the object*

The following measurements were made to detect the influence to the RGB-values, by different distances between sensor and the color. The distance which was normally 2mm we increased to 4mm.

![Graph showing difference by the Red-Led](image1)

![Graph showing difference by the Green-LED](image2)
The graphs are showing the results by 2mm distance and by 4mm distance for each LED. On the x axes are the 24 standard colors, but for each color we took 5 measurements.
In this diagrams we can see, that the distance to the color, which we want to measure is important, because the values of the three LED are changing because of the distance.

**b) Measures with different angel of the sensor**

The following measurements were made to detect the influence to the RGB-values, by different Angle between sensor and object. We had fixed the sensor, with an auxiliary construction, in three different angels (85°- / 85°+ / 90°). The following graphs descript the 24 measured colors.
The curve $85^\circ +$ describes the measurements, when the blue LED is in the origin and the curve $85^\circ -$ describe the measurements, when the red-LED is in the origin. The curve $90^\circ$ describes the measurements, when the sensor is justified on the color. We think, for correct measurement it is important that the angle between the sensor and the object (color) is always the same. So we must show, that the angel is nearly $90^\circ$, because it is easier to holt the sensor with a angel of $90^\circ$ then with $85^\circ+$ or $85^\circ-$.

c) **Temperatures measurement**

We had problems because we do not get repeatable measurements. So we made some thoughts about the temperature. We made measurements to find out if a changing of the temperature of the sensor influences a changing of the RGB Values.

So we made 1000 measurements on a white paper without break between the measurements. The results are held on in the following graph.
On this point we can not say that the changing of the values is an effect of the temperature. So we made new measurements.

At first we made 30 measurements and between the measurements we turned all three Leds on for 2 minutes. On the graph below are the results of this measurement.

The brake of two minutes causes that the sensor can cool down. The values are nearly constant, so the temperature has an influence to the values rather to the Pin Diodes.

**d) Temperatures measurement of the monitor diode**

After a consultation with the producer of the sensor we get an idea how we can solve the problem with the temperature. The sensor has inside a Monitor Diode.
With this Monitor Diode we can measure the temperature, and we can adjust the RGB values.

In the next graphs we made different measurements including the measurements with the Monitor Diode (temperature).
RGB-values of the third measurement

<table>
<thead>
<tr>
<th>Value of the Color</th>
<th>Red</th>
<th>Green</th>
<th>Blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>185,000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>190,000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>195,000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200,000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>205,000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>210,000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Quantity of measures: 11/28
The voltage of the monitor diode depends on the temperature of the sensor.
4. Researches of methods for color identification

An important point is to give out the right color, but at first we have to detect it. To find the right color we use a table with the RGB values of the 24 most important colors.

To explain the methods for detecting the color we take random values for the measured RGB values.

For example:

<table>
<thead>
<tr>
<th>measured value</th>
<th>R</th>
<th>G</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>35</td>
<td>100</td>
<td>70</td>
</tr>
</tbody>
</table>

This example shows you the color bluish green with the index 6. The next step is to calculate the difference between the measured values and the reference values which is pictured below.

<table>
<thead>
<tr>
<th>Reference value</th>
<th>R</th>
<th>G</th>
<th>B</th>
<th>Index</th>
<th>R</th>
<th>G</th>
<th>B</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>dark skin</td>
<td>43,009</td>
<td>26,224</td>
<td>15,589</td>
<td>1</td>
<td>8,009</td>
<td>73,776</td>
<td>54,411</td>
<td>1</td>
</tr>
<tr>
<td>light skin</td>
<td>147,376</td>
<td>75,147</td>
<td>55,27</td>
<td>2</td>
<td>112,376</td>
<td>24,853</td>
<td>14,73</td>
<td>2</td>
</tr>
<tr>
<td>blue sky</td>
<td>36,181</td>
<td>39,887</td>
<td>85,321</td>
<td>3</td>
<td>1,181</td>
<td>60,113</td>
<td>15,321</td>
<td>3</td>
</tr>
<tr>
<td>foliage</td>
<td>27,294</td>
<td>33,864</td>
<td>15,758</td>
<td>4</td>
<td>7,706</td>
<td>66,136</td>
<td>54,242</td>
<td>4</td>
</tr>
<tr>
<td>blue flower</td>
<td>70,648</td>
<td>49,911</td>
<td>109,701</td>
<td>5</td>
<td>35,648</td>
<td>50,089</td>
<td>39,701</td>
<td>5</td>
</tr>
<tr>
<td>bluish green</td>
<td>49,265</td>
<td>101,815</td>
<td>98,12</td>
<td>6</td>
<td>14,265</td>
<td>1,815</td>
<td>28,12</td>
<td>6</td>
</tr>
<tr>
<td>orange</td>
<td>148,275</td>
<td>86,562</td>
<td>13,905</td>
<td>7</td>
<td>112,376</td>
<td>24,853</td>
<td>14,73</td>
<td>7</td>
</tr>
<tr>
<td>purplish blue</td>
<td>22,561</td>
<td>21,964</td>
<td>120,224</td>
<td>8</td>
<td>12,439</td>
<td>78,036</td>
<td>32,041</td>
<td>8</td>
</tr>
<tr>
<td>moderate red</td>
<td>150,229</td>
<td>28,028</td>
<td>33,018</td>
<td>9</td>
<td>115,229</td>
<td>71,972</td>
<td>36,982</td>
<td>9</td>
</tr>
<tr>
<td>purple</td>
<td>35,495</td>
<td>13,254</td>
<td>35,623</td>
<td>10</td>
<td>0,495</td>
<td>86,746</td>
<td>34,377</td>
<td>10</td>
</tr>
<tr>
<td>yellow green</td>
<td>79,791</td>
<td>129,208</td>
<td>19,224</td>
<td>11</td>
<td>44,791</td>
<td>29,208</td>
<td>50,776</td>
<td>11</td>
</tr>
<tr>
<td>orange yellow</td>
<td>164,194</td>
<td>133,479</td>
<td>16,922</td>
<td>12</td>
<td>129,194</td>
<td>33,479</td>
<td>53,078</td>
<td>12</td>
</tr>
<tr>
<td>blue</td>
<td>10,26</td>
<td>9,849</td>
<td>79,661</td>
<td>13</td>
<td>24,74</td>
<td>90,151</td>
<td>9,661</td>
<td>13</td>
</tr>
<tr>
<td>green</td>
<td>23,519</td>
<td>58,385</td>
<td>16,796</td>
<td>14</td>
<td>11,481</td>
<td>41,615</td>
<td>53,204</td>
<td>14</td>
</tr>
<tr>
<td>red</td>
<td>166,618</td>
<td>14,802</td>
<td>11,86</td>
<td>15</td>
<td>131,618</td>
<td>85,198</td>
<td>58,14</td>
<td>15</td>
</tr>
<tr>
<td>yellow</td>
<td>194,183</td>
<td>177,114</td>
<td>15,151</td>
<td>16</td>
<td>159,183</td>
<td>77,114</td>
<td>54,849</td>
<td>16</td>
</tr>
<tr>
<td>magenta</td>
<td>174,562</td>
<td>27,858</td>
<td>78,173</td>
<td>17</td>
<td>139,562</td>
<td>72,142</td>
<td>8,173</td>
<td>17</td>
</tr>
<tr>
<td>cyan</td>
<td>18,601</td>
<td>29,882</td>
<td>90,129</td>
<td>18</td>
<td>16,399</td>
<td>70,118</td>
<td>20,129</td>
<td>18</td>
</tr>
<tr>
<td>white 9.5 (.05 D)</td>
<td>235,122</td>
<td>233,511</td>
<td>226,297</td>
<td>19</td>
<td>200,122</td>
<td>133,511</td>
<td>156,297</td>
<td>19</td>
</tr>
<tr>
<td>neutral 8 (.23 D)</td>
<td>147,211</td>
<td>150,291</td>
<td>150,387</td>
<td>20</td>
<td>112,211</td>
<td>50,291</td>
<td>80,387</td>
<td>20</td>
</tr>
<tr>
<td>neutral 6.5 (.44 D)</td>
<td>89,186</td>
<td>91,952</td>
<td>92,186</td>
<td>21</td>
<td>54,186</td>
<td>8,048</td>
<td>22,484</td>
<td>21</td>
</tr>
<tr>
<td>neutral 5 (.70 D)</td>
<td>47,088</td>
<td>49,874</td>
<td>49,874</td>
<td>22</td>
<td>12,088</td>
<td>50,726</td>
<td>20,122</td>
<td>22</td>
</tr>
<tr>
<td>neutral 3.5 (1.05 D)</td>
<td>21,758</td>
<td>22,841</td>
<td>23,363</td>
<td>23</td>
<td>13,242</td>
<td>77,159</td>
<td>46,637</td>
<td>23</td>
</tr>
<tr>
<td>black 2 (1.5 D)</td>
<td>8,09</td>
<td>8,105</td>
<td>8,311</td>
<td>24</td>
<td>26,91</td>
<td>91,895</td>
<td>61,689</td>
<td>24</td>
</tr>
</tbody>
</table>
a) First method

Now we take the 12 minimum values of the red-LED and accept the two other values form the two other colors.

<table>
<thead>
<tr>
<th>Index</th>
<th>12 min of red value</th>
<th>accept</th>
<th>accept</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.495</td>
<td>86.746</td>
<td>34.377</td>
</tr>
<tr>
<td>3</td>
<td>1.181</td>
<td>60.113</td>
<td>15.321</td>
</tr>
<tr>
<td>4</td>
<td>7.706</td>
<td>66.136</td>
<td>54.242</td>
</tr>
<tr>
<td>1</td>
<td>8.009</td>
<td>73.776</td>
<td>54.411</td>
</tr>
<tr>
<td>14</td>
<td>11.481</td>
<td>41.615</td>
<td>53.204</td>
</tr>
<tr>
<td>22</td>
<td>12.088</td>
<td>50.726</td>
<td>20.122</td>
</tr>
<tr>
<td>8</td>
<td>12.439</td>
<td>78.036</td>
<td>32.041</td>
</tr>
<tr>
<td>23</td>
<td>13.242</td>
<td>77.159</td>
<td>46.637</td>
</tr>
<tr>
<td>6</td>
<td>14.265</td>
<td>1.815</td>
<td>28.120</td>
</tr>
<tr>
<td>18</td>
<td>16.399</td>
<td>70.118</td>
<td>20.129</td>
</tr>
<tr>
<td>13</td>
<td>24.740</td>
<td>90.151</td>
<td>9.661</td>
</tr>
<tr>
<td>24</td>
<td>26.910</td>
<td>91.895</td>
<td>61.689</td>
</tr>
</tbody>
</table>

In the second step, we pick out the 6 Minimum values of the green-LED from the table on the top.

<table>
<thead>
<tr>
<th>Index</th>
<th>accept</th>
<th>6 min. of green value</th>
<th>accept</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>14,265</td>
<td>1,815</td>
<td>28,120</td>
</tr>
<tr>
<td>14</td>
<td>11,481</td>
<td>41,615</td>
<td>53,204</td>
</tr>
<tr>
<td>22</td>
<td>12,088</td>
<td>50,726</td>
<td>20,122</td>
</tr>
<tr>
<td>3</td>
<td>1,181</td>
<td>60,113</td>
<td>15,321</td>
</tr>
<tr>
<td>4</td>
<td>7,706</td>
<td>66,136</td>
<td>54,242</td>
</tr>
<tr>
<td>18</td>
<td>16,399</td>
<td>70,118</td>
<td>20,129</td>
</tr>
</tbody>
</table>

In the third step, we pick out the 3 minimum values of the blue-LED from the table above.

<table>
<thead>
<tr>
<th>Index</th>
<th>accept</th>
<th>accept</th>
<th>3 min. of blue value</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1,181</td>
<td>60,1125408</td>
<td>15,321</td>
</tr>
<tr>
<td>22</td>
<td>12,087</td>
<td>50,7264448</td>
<td>20,122</td>
</tr>
<tr>
<td>18</td>
<td>16,399</td>
<td>70,1175488</td>
<td>20,129</td>
</tr>
</tbody>
</table>

Now we made the sum of all values of each index. The result is:

<table>
<thead>
<tr>
<th>Index</th>
<th>absolute deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>76,615</td>
</tr>
<tr>
<td>22</td>
<td>82,936</td>
</tr>
<tr>
<td>18</td>
<td>106,646</td>
</tr>
</tbody>
</table>

The result with the first method is the color **blue sky** with the index 3.
b) Second method

In the second method we used an allowance for the research, in this we choose index 10. This means, that the measured values can have only a difference of +/-10 to the reference value.

In the first step we pick out the values from each LED, where the difference is lower +/- 10 and made the sum of all values of each index

<table>
<thead>
<tr>
<th>Red-LED</th>
<th>sum of the differences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>R</td>
</tr>
<tr>
<td></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Green-LED</th>
<th>sum of the differences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>R</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Blue-LED</th>
<th>sum of the differences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>R</td>
</tr>
<tr>
<td></td>
<td>13</td>
</tr>
</tbody>
</table>

The result with the second method is the color **bluish green** with the index 6.

c) Third method

The third method is nearly the same like the second method. The difference is only, that we don’t make the sum of all differences. We make a weighting of all values from each index.

d) Comparison of the three methods

Comparison of the three methods

<table>
<thead>
<tr>
<th>Method</th>
<th>memory capacity</th>
<th>computing power</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dependent on the accuracy, the smaller the range of tolerance, the smaller is the storage requirement. However it can occur with the fact that the ‘correct’ color is excluded.</td>
<td>This will need surely at few computing power</td>
</tr>
<tr>
<td>2</td>
<td>Depend on the allowance, the smaller it is, the smaller is the storage requirement.</td>
<td>This will need more computing power because we use many values which are needed for the result.</td>
</tr>
</tbody>
</table>
As a conclusion, we can say, all the three methods have a possibility, to get the false color-name. So we have found an easier and exacting method to find the correct color.

**e) Last method for color identification**

The method which we are using to detect the color is a mathematical method. With Red, Green and Blue we can create a RGB - Color room, in this room counts the same mathematical rules like in other vector rooms.

To detect the color we calculate the smallest vector to the reference colors. This is a normal vector calculation.

Calculation:

\[
\begin{align*}
\text{smallest vector...} v &= \sqrt{(rv - mrv)^2 + (gv - mgv)^2 + (bv - mbv)^2} \\
\end{align*}
\]

For every reference color we get a result, but the smallest value gives as the correct color.
5. Selection of the construction units - Hardware

a) Introduction

The Hardware is an important component of this project. It is divided in three main parts: Microcontroller, Sensor, Speech part. Each main part needs other parts and components for the correct functionality.

b) Power supply

A 3 Volt Battery is used for the power supply. The Microcontroller and the Speech Chip needs 5 Volt, so the 3 Volt is not enough. The MAX756 is a step up converter which converts an input voltage between 1.7 Volt to 5 Volt up to 3 Volt or to 5 Volt. This Chip detects also when the battery is low. With 5 Volt output voltage it is possible to get an output current up to 200 mA.

![Diagram of step up converter]

The picture shows the Layout of the step up converter.

c) Sensor

The Sensor consists of a Red LED, Green LED, Blue LED, two Photo Diodes, and a Monitor Diode. There is also a connected ground and a reference Voltage for the Photo Diodes.

The Red LED needs a voltage around 1.91 Volt and a current of 14mA, so the resistor between microcontroller and the LED must be 22K. The blue LED needs a voltage of 3.75 Volt and a current of 7mA so the resistor must be 180Ohms. The Green LED takes 5 Volt and a current of 32mA, but the microcontroller has an output current of a maximum of 25mA. So the current is given by a transistor. The basis of this transistor is connected to the microcontroller and the emitter is connected to the green LED. If
the output port of the microcontroller is set low the transistor locks. If the output port is high the green LED gets the current. The Photo Diodes are used for detecting the light intensity. The resistor or the conductivity of the Photo Diodes is changing when the light intensity is changing. The Monitor Diode is used for detecting the temperature in the sensor. It is also changing the resistor when the temperature is changing.

d) The Amplifier

An amplifier is used to amplify the signal of the Photodiodes. The amplifier is an inverting amplifier. It consists of an OPV LM358 a resistor and a capacitor. If the resistor of the Photo Diodes is changing the output voltage of the OPV is also changing. The capacitor is used to stable the voltage.

e) Microcontroller

We use the PIC18F2455 from microchip. More information about the microcontroller please visit the datasheet (http://www.microchip.com/downloads/en/DeviceDoc/39632c.pdf). The Voltage of the Pin Diode1 is connected with the Pin 2 (Port A0), and the Pin Diode 2 is connected with Pin 3 (Port A1). On the Pin3 we find the Monitor Diode. On the Pins 11, 12, 13 are connected the Leds (Blue, Red, Green). It is possible to detect the low battery with the Port RC5 (Pin16). On Pin 18 (Port RC7) is the bush bottom to start a measurement. The Pins 21, 22, 23, 24, 25 (Port B0…B4) are the address lines for the Speech Chip. With Pin 27 (Port B6) we can detect when a played message is finish. To start playing a message we give the signal on Pin 28 (Port B7).
f) **Speech chip**

We are using the speech chip to give out the name of the color by a speaker. After researches we have decided to use an ISD chip, because it was the only chip which is able to give out an analog signal.

For us it was important because the digital output of a voice is very hard to understand.

We are using the ISD 1420, which can store 20 seconds. The following graph shows the protective circuit.

![Diagram of ISD 1420 protective circuit](image)

This circuit was important to record the different messages (color names). The addresses A0-A7 we are using to select the position of the message with a binary code. The buttons PLAYL / PLAYE / REC are to play or store the messages. The AGC input is for amplify of the input signal. The input signal can be transmitted by two different ways. The first way we are using is a Microphone and the second way we set the output “ANA OUT” after the capacitor at Ground. Then we connect an analog signal (PC/ MP3-player …) on the “ANA IN”.

For the output we are using an 8 Ohm speaker.
g) Schematic

![Schematic Diagram]

h) Board

![Board Diagram]
6. The design
7. Introduction of Software Development

The goal of this report is to development of software for microcontroller to develop the embedded system of the color detection project. The basic idea of the software is to communicate with two modules of the embedded system sensor and speech chip.

First of all, Ports configuration done of microcontroller. After that it turns on each LED of the sensor in sequence and when the Led is turning on, Pin diodes of the sensor measures the reflection of Led light and it was simply done by calling analog to digital function (ADC).

Then some mathematical operations are come in the picture, which are applied on result of ADC to make it comparable with universal RGB values (stored in arrays).

Next step is to compare this measured and calculated RGB values with universal RGB values and here, software finds the minimum distance between measured and universal RGB values.

At the end, after finding minimum distance it is ready to send 5 bits address to speech chip where the colours are stored and after receiving this message speech chip is responsible to speak the matching color.

The microcontroller used was PIC18F2455 from microchip. The programmer tools were MPLAB and PICSTART Plus from Microchip. All the programming development was made in ANSI C using Microchip C18 compiler.

For more information complete software implementation is included.
a) The Program

```c
#include <stdio.h>
#include <math.h>
#include <adc.h>
#include <delays.h>
#include <p18f2455.h>

//Function declaration
void color_detection(int, int, int);
void play(void);
int adc_1(void);
int adc_2(void);

//Global arrays
unsigned long red[] = {43, 147, 36, 27, 71, 49, 148, 23, 35, 164, 10, 24, 167, 194, 175, 19, 235, 47, 8};
unsigned long green[] = {26, 75, 40, 34, 50, 102, 87, 22, 13, 133, 10, 58, 15, 177, 28, 30, 234, 49, 8};
unsigned long blue[] = {16, 55, 85, 16, 110, 98, 14, 102, 36, 17, 80, 17, 12, 15, 78, 90, 226, 50, 8};
unsigned long new[19], search=0;

void main (void)
{
    int voltage_r, voltage_g, voltage_b;
    int voltage1,voltage2,voltage3,voltage_r,voltage_g,voltage_b;
    unsigned long i=0;
    unsigned long delay=65000;

    //TRIS register configuration
    TRISA=0xFF;// Set PORTA pins as input
    TRISB=0x40;// Set PORTB <0.3>pins as output and <4.7>pins as input
    TRISC=0x80;// Set PORTC <7>pin as input rest are outputs
    PORTC=0x00;// Clear PORTC
    PORTB=0x00;// Clear PORTB
    PORTCbits.RC7=1;//Set push button to high
    PORTBbits.RB7=1;//Set digital output on high for playing message

    OpenADC(ADC_FOSC_32 & ADC_LEFT_JUST , ADC_CH0 & ADC_CH1 & ADC_16_TAD & ADC_INT_OFF &
    ADC_VREFPLUS_VDD & ADC_VREFMINUS_VSS , 13);//Initialize analog to digital converter
    while(1)
    {
        if(PORTCbits.RC7==0)//Wait for press button to set on low
        {
            for(i=0;i<1000;i++)//Turn on Red LED for short time
            {
                PORTC=0x02;
                voltage_r=adc_1();// Store the result of Analog to Digital conversion(Red LED)
                voltage_r=adc_2();
            }
            voltageg=(unsigned int)((voltage_r+voltage_r)/2);
            //Next two instructions converts the result(Red LED) into RGB form
            voltage1= (voltageg-180)*(-1);
            voltage_g = (voltage1*3);  
            i = 0;
            for(i=0;i<delay;i++)//Turn off Red LED and wait for short time
            {
                PORTC=0x00;
            }
        }
        for(i=0;i<1000;i++)//Turn on Green LED for short time
        {
            PORTC=0x04;
            voltage_g=adc_1();// Store the result of Analog to Digital conversion(Green LED)
            voltage_g=adc_2();
            i=0;
            voltageg=(unsigned int)((voltage_g+voltage_g)/2);
            //Next two instructions converts the result(Green LED) into RGB form
            voltage2= (voltageg-180)*(-1);
            voltage_g = (voltage2*7);  
            for(i=0;i<delay;i++)//Turn off Green LED and wait for short time
            {
                PORTC=0x00;
            }
        }
    }
}
```
for(i=0;i<1000;i++)//Turn on Blue LED for short time
{
    PORTC=0x01;// Store the result of Analog to Digital conversion(Blue LED)
    voltage_b1=adc_1();
    voltage_b2=adc_2();
    i=0;
    voltageb=(unsigned int)((voltage_b1+voltage_b2)/2);//Next two instructions converts the result(Blue LED) into RGB form
    voltage3= (voltageb-180)*(-1);
    voltage_b = (voltage3*4);
    for(i=0;i<delay;i++);//Turn off Blue LED and wait for short time
    PORTC=0x00;
    i=0;
    PORTCb.RC7=0;//Set again push button to high
    color_detection(voltage_r, voltage_g, voltage_b);//Call the color_detection function
}
}
CloseADC();

//This function is responsible for detecting the color and sending the message to Speech chip
void color_detection(int voltage_r, int voltage_g, int voltage_b)
{
    unsigned long nred=0, ngreen=0, nblue=0, i=0;
    //This loop finds the distances between measured colors result and universal RGB color values
    for(i=0;i<=18;i++)
    {
        nred= (unsigned long)((red[i]-voltage_r)*(red[i]-voltage_r));
        ngreen= (unsigned long)((green[i]- voltage_g)*(green[i] - voltage_g));
        nblue= (unsigned long)((blue[i] - voltage_b)*(blue[i] - voltage_b));
        new[i]= (unsigned long) (nred + ngreen + nblue);
    }
    i=0;
    search=(unsigned long)new[0];
    //This loop search for minimum distance
    for(i=0;i<18;i++)
    {
        if(search>new[i+1])
        {
            search=new[i+1];//printf("%lu
",search);
        }
    }
    //This loop compares the minimum distance with already stored distances and also pass the suitable 5 bits message to
    Speech chip.
    if(new[0]==search)
    {
        PORTBbits.RB0=0;
        PORTBbits.RB1=0;
        PORTBbits.RB2=0;
        PORTBbits.RB3=0;
        PORTBbits.RB4=0;
        play();// Call the function play
    }
    else if(new[1]==search)
    {
        PORTBbits.RB0=1;
        PORTBbits.RB1=0;
        PORTBbits.RB2=0;
        PORTBbits.RB3=0;
        PORTBbits.RB4=0;
        play();
    }
    else if(new[2]==search)
    {
        PORTBbits.RB0=0;
        PORTBbits.RB1=1;
        PORTBbits.RB2=0;
        PORTBbits.RB3=0;
        PORTBbits.RB4=0;
        play();
    }
    else if(new[3]==search)
if(new[4]==search) {
    PORTBbits.RB0=0;
    PORTBbits.RB1=0;
    PORTBbits.RB2=1;
    PORTBbits.RB3=0;
    PORTBbits.RB4=0;
    play();
}
else if(new[5]==search) {
    PORTBbits.RB0=0;
    PORTBbits.RB1=0;
    PORTBbits.RB2=1;
    PORTBbits.RB3=0;
    PORTBbits.RB4=0;
    play();
}
else if(new[6]==search) {
    PORTBbits.RB0=0;
    PORTBbits.RB1=1;
    PORTBbits.RB2=1;
    PORTBbits.RB3=0;
    PORTBbits.RB4=0;
    play();
}
else if(new[7]==search) {
    PORTBbits.RB0=0;
    PORTBbits.RB1=0;
    PORTBbits.RB2=1;
    PORTBbits.RB3=0;
    PORTBbits.RB4=0;
    play();
}
else if(new[8]==search) {
    PORTBbits.RB0=0;
    PORTBbits.RB1=1;
    PORTBbits.RB2=0;
    PORTBbits.RB3=1;
    PORTBbits.RB4=0;
    play();
}
else if(new[9]==search) {
    PORTBbits.RB0=0;
    PORTBbits.RB1=0;
    PORTBbits.RB2=0;
    PORTBbits.RB3=1;
    PORTBbits.RB4=0;
    play();
}
else if(new[10]==search) {
    PORTBbits.RB0=0;
    PORTBbits.RB1=1;
    PORTBbits.RB2=0;
    PORTBbits.RB3=1;
    PORTBbits.RB4=0;
    play();
}
else if(new[11]==search) {
    PORTBbits.RB0=0;
    PORTBbits.RB1=1;
    PORTBbits.RB2=0;
    PORTBbits.RB3=1;
    PORTBbits.RB4=0;
    play();
}
PORTBbits.RB4=0;
play();
} else if(new[12]==search)
{
PORTBbits.RB0=0;
PORTBbits.RB1=0;
PORTBbits.RB2=1;
PORTBbits.RB3=1;
PORTBbits.RB4=0;
play();
} else if(new[13]==search)
{
PORTBbits.RB0=1;
PORTBbits.RB1=0;
PORTBbits.RB2=1;
PORTBbits.RB3=1;
PORTBbits.RB4=0;
play();
} else if(new[14]==search)
{
PORTBbits.RB0=0;
PORTBbits.RB1=1;
PORTBbits.RB2=1;
PORTBbits.RB3=1;
PORTBbits.RB4=0;
play();
} else if(new[15]==search)
{
PORTBbits.RB0=1;
PORTBbits.RB1=1;
PORTBbits.RB2=1;
PORTBbits.RB3=1;
PORTBbits.RB4=0;
play();
} else if(new[16]==search)
{
PORTBbits.RB0=0;
PORTBbits.RB1=0;
PORTBbits.RB2=0;
PORTBbits.RB3=0;
PORTBbits.RB4=1;
play();
} else if(new[17]==search)
{
PORTBbits.RB0=1;
PORTBbits.RB1=0;
PORTBbits.RB2=0;
PORTBbits.RB3=0;
PORTBbits.RB4=1;
play();
} else if(new[18]==search)
{
PORTBbits.RB0=0;
PORTBbits.RB1=1;
PORTBbits.RB2=0;
PORTBbits.RB3=0;
PORTBbits.RB4=1;
play();
}

//Here analog to digital conversion happens
int adc_1(void)
{
  int result;
  SetChanADC(ADC_CH0);
  Delay10TCYx(5);//Wait for 50 instruction cycle
  ConvertADC();//Start A/D converter
  while(BusyADC());//wait until ADC is busy
  ReadADC();//Read analog to digital conversion
int adc_2(void) {
    int result;
    SetChanADC(ADC_CH1);
    Delay10TCYx(5); // Wait for 50 instruction cycle
    ConvertADC(); // Start A/D converter
    while(BusyADC()); // Wait until ADC is busy
    result = ADRESH; // Store the 8 bits result
    return(result);
}

void play(void) {
    int i = 0;
    for(i=0;i<2;i++) {
        PORTBbits.RB7 = 0;
    }
    PORTBbits.RB7 = 1;
}