**CODING**

1. RM 1.00 ;

 if ((v1 > 450 && v2 > 400 && v3 > 700) &&( v1 < 900 && v2 < 900 && v3 < 1500 ))

 { Serial.println ("This is Rm 1");

 mp3\_play(1);

 delay (2000);

 }

1. RM 10.00 ;

else if ((v1 > 1000 && v2 > 400 && v3 > 600) &&( v1 < 1900 && v2 < 1000 && v3 < 1100 ))

 {Serial.println ("This is Rm 10");

 mp3\_play(3);

1. RM 20.00 ;

else if ((v1 > 1000 && v2 > 800 && v3 > 700) &&( v1 < 2000 && v2 < 950 && v3 < 980 ))

 {Serial.println ("This is Rm 20");

 mp3\_play(2);

 delay (2000);}

1. RM 50.00 ;

else if((v1 > 1010 && v2 > 1100 && v3 > 1700) &&( v1 < 1500 && v2 < 1490 && v3 < 2190 ))

 {Serial.println ("This is Rm 50");

 mp3\_play(5);

 delay (2000);}

1. RM 100.00

else if ((v1 > 400 && v2 > 300 && v3 > 650) &&( v1 < 600 && v2 < 500 && v3 < 800))

 {Serial.println ("This is Rm 100");

 mp3\_play(6);

 delay (2000);

 }

1. RM 5.00

else if ((v1 > 800 && v2 > 800 && v3 > 900) &&( v1 < 1300 && v2 < 1400 && v3 < 1500 ))

 {Serial.println ("This is Rm 5");

 mp3\_play(4);

 delay (2000);

 }

#include <DFPlayer\_Mini\_Mp3.h>

#include <SoftwareSerial.h>

#include <TimerOne.h>

SoftwareSerial mySerial (9,8); //RX,TX

#define S0 6

#define S1 5

#define S2 4

#define S3 3

#define OUT 2

// OE to GND

int g\_count = 0; // count the frequecy

int g\_array[3]; // store the RGB value

int g\_flag = 0; // filter of RGB queue

float g\_SF[3]; // save the RGB Scale factor

int v1, v2, v3;

// Init TSC230 and setting Frequency.

void TSC\_Init()

{

 pinMode(S0, OUTPUT);

 pinMode(S1, OUTPUT);

 pinMode(S2, OUTPUT);

 pinMode(S3, OUTPUT);

 pinMode(OUT, INPUT);

 digitalWrite(S0, LOW); // OUTPUT FREQUENCY SCALING 2%

 digitalWrite(S1, HIGH);

}

// Select the filter color

void TSC\_FilterColor(int Level01, int Level02)

{

 if(Level01 != 0)

 Level01 = HIGH;

 if(Level02 != 0)

 Level02 = HIGH;

 digitalWrite(S2, Level01);

 digitalWrite(S3, Level02);

}

void TSC\_Count()

{

 g\_count ++ ;

}

void TSC\_Callback()

{

 switch(g\_flag)

 {

 case 0:

 //Serial.println("->WB Start");

 TSC\_WB(LOW, LOW); //Filter without Red

 break;

 case 1:

 //Serial.print("->Frequency R=");

 //Serial.println(g\_count);

 g\_array[0] = g\_count;

 TSC\_WB(HIGH, HIGH); //Filter without Green

 break;

 case 2:

 //Serial.print("->Frequency G=");

 //Serial.println(g\_count);

 g\_array[1] = g\_count;

 TSC\_WB(LOW, HIGH); //Filter without Blue

 break;

 case 3:

 //Serial.print("->Frequency B=");

 //Serial.println(g\_count);

 //Serial.println("->WB End");

 g\_array[2] = g\_count;

 TSC\_WB(HIGH, LOW); //Clear(no filter)

 break;

 default:

 g\_count = 0;

 break;

 }

}

void TSC\_WB(int Level0, int Level1) //White Balance

{

 g\_count = 0;

 g\_flag ++;

 TSC\_FilterColor(Level0, Level1);

 Timer1.setPeriod(1000000); // set 1s period

}

void setup()

{

 mySerial.begin(9600);

 Serial.begin(57600);

 mp3\_set\_serial (mySerial);

 delay (1);

 mp3\_set\_volume (30);

 TSC\_Init();

 Timer1.initialize(); // defaulte is 1s

 Timer1.attachInterrupt(TSC\_Callback);

 attachInterrupt(0, TSC\_Count, RISING);

 delay(4000); // Dont change this

 for(int i=0; i<3; i++)

 //Serial.println(g\_array[i]);

 g\_SF[0] = 255.0/ g\_array[0]; //R Scale factor

 g\_SF[1] = 255.0/ g\_array[1] ; //G Scale factor

 g\_SF[2] = 255.0/ g\_array[2] ; //B Scale factor

 //Serial.println(g\_SF[0]);

 //Serial.println(g\_SF[1]);

 //Serial.println(g\_SF[2]);

}

void loop()

{

 g\_flag = 0;

 for(int i=0; i<3; i++)

 {

 //Serial.println(int(g\_array[i] \* g\_SF[i]));

 v1 = (g\_array[0]);

 v2 = (g\_array[1]);

 v3 = (g\_array[2]);

 }

 Serial.print ("v1: ");

 Serial.println (v1);

 Serial.print ("v2: ");

 Serial.println (v2);

 Serial.print ("v3: ");

 Serial.println (v3);

 if ((v1 > 450 && v2 > 400 && v3 > 700) &&( v1 < 900 && v2 < 900 && v3 < 1500 ))

 { Serial.println ("This is Rm 1");

 mp3\_play(1);

 delay (2000);

 }

 else if ((v1 > 400 && v2 > 300 && v3 > 650) &&( v1 < 600 && v2 < 500 && v3 < 800 ))

 {Serial.println ("This is Rm 100");

 mp3\_play(6);

 delay (2000);

 }

 else if ((v1 > 1000 && v2 > 800 && v3 > 700) &&( v1 < 2000 && v2 < 950 && v3 < 980 ))

 {Serial.println ("This is Rm 20");

 mp3\_play(2);

 delay (2000);}

 else if ((v1 > 1000 && v2 > 400 && v3 > 600) &&( v1 < 1900 && v2 < 1000 && v3 < 1100 ))

 {Serial.println ("This is Rm 10");

 mp3\_play(3);

 delay (2000);}

 else if((v1 > 1010 && v2 > 1100 && v3 > 1700) &&( v1 < 1500 && v2 < 1490 && v3 < 2190 ))

 {Serial.println ("This is Rm 50");

 mp3\_play(5);

 delay (2000);}

 else if ((v1 > 800 && v2 > 800 && v3 > 900) &&( v1 < 1300 && v2 < 1400 && v3 < 1500 ))

 {Serial.println ("This is Rm 5");

 mp3\_play(4);

 delay (2000);

 }

 else

 Serial.println ("undetected");

 delay(4000); // delay so its give a little bit time to user