#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd (0x27, 20, 4);

short LEDS [] = {8, 9, 10, 11};

short Vect[] = {2, 3, 4, 5, 6};

void S(short int ite, bool Out1 = HIGH, bool Out2 = LOW)

{

  digitalWrite(Vect[ite], Out1);

  digitalWrite(Vect[ite+1], Out2);

  delay(200);

}

void show (const char\* str, const int x = 0, const int y = 0, const bool clear = false) {

    if (clear)

        lcd.clear();

    if (x >= 0 && y >= 0)

        lcd.setCursor(x, y);

    lcd.print(str);

}

void setup()

{

  Serial.begin(9600);

  lcd.init();

  lcd.backlight();

    for(short i = 0; i < 4; i++){

      if(Vect[i] != 6)

        pinMode(Vect[i], OUTPUT);

      else

        pinMode(Vect[i], INPUT);

    }

   for(short i = 0; i < 4; i++)

        pinMode(LEDS[i], OUTPUT);

  S(0);

}

void loop()

{

  for(short i = 0; i < 4; i++)

       digitalWrite(LEDS[i], LOW);

  S(2, LOW);

  int rojo = pulseIn(Vect[4], LOW);

  S(2, HIGH, HIGH);

  int verde = pulseIn(Vect[4], LOW);

  S(2, LOW, HIGH);

  int azul = pulseIn(Vect[4], LOW);

  //naranja

  if(rojo < 215 && (verde > 350 && verde < 380) && (azul > 300 && azul < 330) ){

    digitalWrite(LEDS[1], HIGH);

    show(" -Naranja- ", 2, 1, true);

  }

  //rojo

  else if((rojo > 230 && rojo < 260) && verde > 390 && azul > 300){

    digitalWrite(LEDS[0], HIGH);

    show(" -Rojo- ", 4, 1, true);

  }

  //verde

  else if(verde < 370 && (rojo > 350 && rojo < 380) && (azul > 325 && azul < 345)){

    digitalWrite(LEDS[2], HIGH);

    show(" -Verde- ", 4, 1, true);

  }

  //azul

  else if(azul < 210 && (verde > 260 && verde < 290) && rojo > 280){

    digitalWrite(LEDS[3], HIGH);

    show(" -Azul- ", 4, 1, true);

  }

  delay(1000);

}

Se declaran las librerías para usar

Se declara el objeto lcd, se hacen 2 arreglos de tipo short (numéricos), para almacenar números los cuales corresponden a ciertos pines y estos tienen funciones

En el arreglo Vect, los pines 2, 3,4,5 son S0, S1, S2 y S3 respectivamente, el pin 6 es la salida de datos del sensor, aquí se tiene que entrar un poco a detalle a como funciona el sensor de colores.  
S0 y S1 se usan para determinar la velocidad de transmisión de datos hacia el Arduino, es decir que esto determina como saca los fatos el sensor de colores, y hay que tener en cuenta que el Arduino no es capaz de interpretar la velocidad completa de este sensor, por lo que se tiene que activar S0 en HIGH y S1 en LOW para que la transmisión de datos tenga una velocidad del 50% de lo normal, el S2 y S3 se utilizan para activar los filtros del sensor, hay foto diodos que tienen filtro s de color azul, otros de color verde otros de color rojo y otros sin filtros, con S2 y S3 se determina que filtro se activa para hacer la lecturas del color.  
Lo que nos interesa es la siguiente tabla para analizar el rango que nos proporciona el verde, azul y rojo con cada color analizado y así determinar un intervalo para cada color y poder encender el led correcto y mostrar el mensaje correcto en la LCD.  
  
Ya después de que se le asigna una variable para la lec5tura de cada filtro, excepto el transparente se analiza en un if else cada intervalo, se le da un intervalo de 1000 segundo para volver a realizar la lectura del sensor, además de que este segundo sirve para poder apreciar que led se prende y el mensaje de la lcd, además de que en cada ciclo loop los LEDS se apagan para evitar tener que almacenar la posición del led que se enciende y luego apagarlo cuando se detecte otro color.