**UNVERSIDAD TÉCNICA DE AMBATO**

**FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL**

**INGENIERÍA EN ELECTRÓNICA Y COMUNICACIONES**



***Practica***

**Tema:**

“Comunicación inalámbrica para un sensor de clasificación de colores "

**Carrera:**

Ingeniería en Electrónica y Comunicaciones.

**Área Académica:**

Física-Electrónica

**Línea de Investigación:**

Nanotecnología

**Ciclo Académico:**

Marzo 2017 - Septiembre 2017

**Paralelo:**

Sexto “A” Electrónica.

**Alumnos:**

Giancarlo Culcay

Freire Juan

Pastuña Henry

**Módulo y Docente:**

Interfaz de PC Ing. Franklin Salazar

Ambato-Ecuador

1. **INFORME PROYECTO**
2. **PP**
   1. **Título**

“Comunicación inalámbrica para un sensor de clasificación de colores "

* 1. **Objetivo General:**

Realizar una comunicación inalámbrica a un sensor de clasificación de colores mediante la utilización de bluetooth para observa la comunicación de diversos comandos.

* 1. **Objetivos Específicos:**
* Realizar una comunicación inalámbrica
* Investigar el método de comunicación a utilizar mediante Arduino.
* Tener la capacidad implementar la comunicación para un buen funcionamiento.
* Conocer la ventaja de realizar las comunicaciones inalámbricas.
  1. **Introducción**

Redes Inalámbricas Como medio físico de transmisión se usan ondas de luz o, principalmente, de radio. Hasta hace poco era una tecnología cara, con muy poca seguridad y que ofrecía velocidades bajas. Los anteriores problemas se superaron a finales de los 90. Las redes inalámbricas son un complemento, no un sustituto, de las redes cableadas. En general, las redes cableadas son más rápidas, seguras y ofrecen mejor rendimiento. Y son más baratas si ya está instalado el cable.

* 1. **Palabras claves**

(Redes Inalámbricas, ondas, transmisión)

* 1. **Resumen**

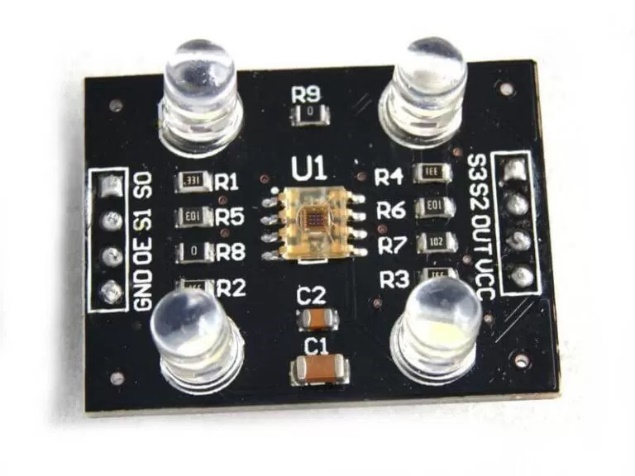
El propósito de esta práctica se basa en realizar una comunicación inalámbrica a un sensor de color a través de la placa Arduino programa que nos permite complicar el código de programación para el desarrollo de la práctica el cual debemos utilizar dos bluetooth para la realización de dicha práctica para ello se investiga y analiza la manera más conveniente en la comunicación para poderlo compartir con el curso Sexto “A” y con todas las personas interesadas en área de las comunicaciones inalámbricas motivos y razones que motivaron la práctica, los objetivos y situación problemática.

* 1. **Materiales y Metodología**
     1. **Materiales**
* Sensor de color TCS3200 con Arduino
* Arduino
* Bluetooth
* Cables
* Resistencias
* Leds
* Materiales para banda transportadora
* Servos
* Motores
* Proto Board
  1. **Marco Teórico**

**Sensor de color TCS3200 con Arduino**

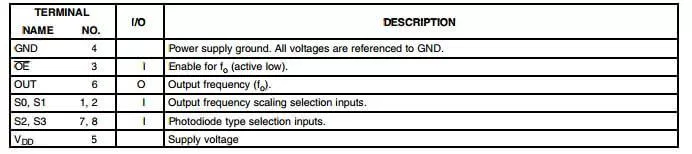
El sensor de color RGB TCS3200 LC Tech es un convertidor de luz a frecuencia que combina fotodiodos de silicio reconfigurables y una corriente de frecuencia en un solo circuito integrado. La salida es una onda cuadrada (ciclo de trabajo 50%) con una frecuencia directamente proporcional a la intensidad de luz. Las entradas y salidas digitales permiten una interfaz directa con un microcontrolador u otro conjunto de circuitos lógicos, por esta razón el sensor TCS3200 es ideal para líneas de producción, domótica, robótica, etc.

Es un convertidor de luz a frecuencia lee una matriz de 8×8 fotodiodos, de tal manera que 16 fotodiodos tienen filtro azul, 16 fotodiodos tienen filtro verde, 16 fotodiodos tienen filtro rojo y 16 fotodiodos son sin filtro. [1]



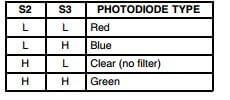
**Cuadro 1: Sensor de color TCS3200 con Arduino**

**FUNCIÓN DE TERMINALES**

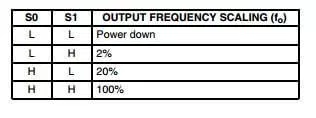


**Cuadro 2: Función de terminales del sensor**

**OPCIONES SELECCIONABLES**

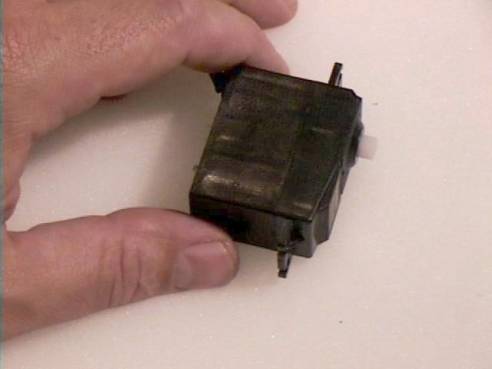
****

**Cuadro3: Tipo de fotodiodo**

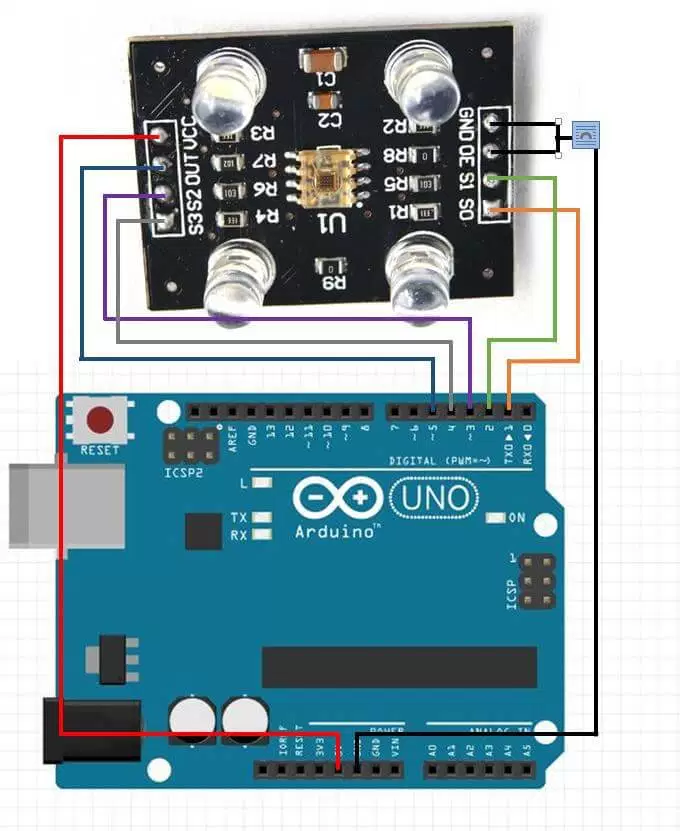


**Cuadro 4: Escala de frecuencia de salida**

**Servomotores**

Los servos son un tipo especial de motor de c.c. que se caracterizan por su capacidad para posicionarse de forma inmediata en cualquier posición dentro de su intervalo de operación. Para ello, el servomotor espera un tren de pulsos que se corresponde con el movimiento a realizar. Están generalmente formados por un amplificador, un motor, un sistema reductor formado por ruedas dentadas y un circuito de realimentación, todo en una misma caja de pequeñas dimensiones. El resultado es un servo de posición con un margen de operación de 180° aproximadamente. [2]

**Cuadro 5: servos**



**Cuadro 6: Diagrama de conexiones**

**Comunicación inalámbrica con Arduino**

Las necesidades para conseguir esto son de lo más variopintas, desde controlar el comportamiento de nuestro Arduino a distancia ya que lo estamos usando para leer datos de un sensor en un lugar apartado del PC, hasta permitir la comunicación inalámbrica de Arduino.

**Zigbee**

En realidad, Zigbee es una especificación de protocolos de comunicación inalámbricos definidos por la *Zigbee Alliance*y enfocados a entornos de bajo consumo. Varios fabricantes producen módulos compatibles, como el que aparece en la siguiente foto:

**[](http://3.bp.blogspot.com/-kUwxXgAB2Go/Us6vKs4AeUI/AAAAAAAABHU/eq5qFN5DkH0/s1600/xbee.jpg)**

**Cuadro 7:Módulo XBee, compatible con el protocolo Zigbee**

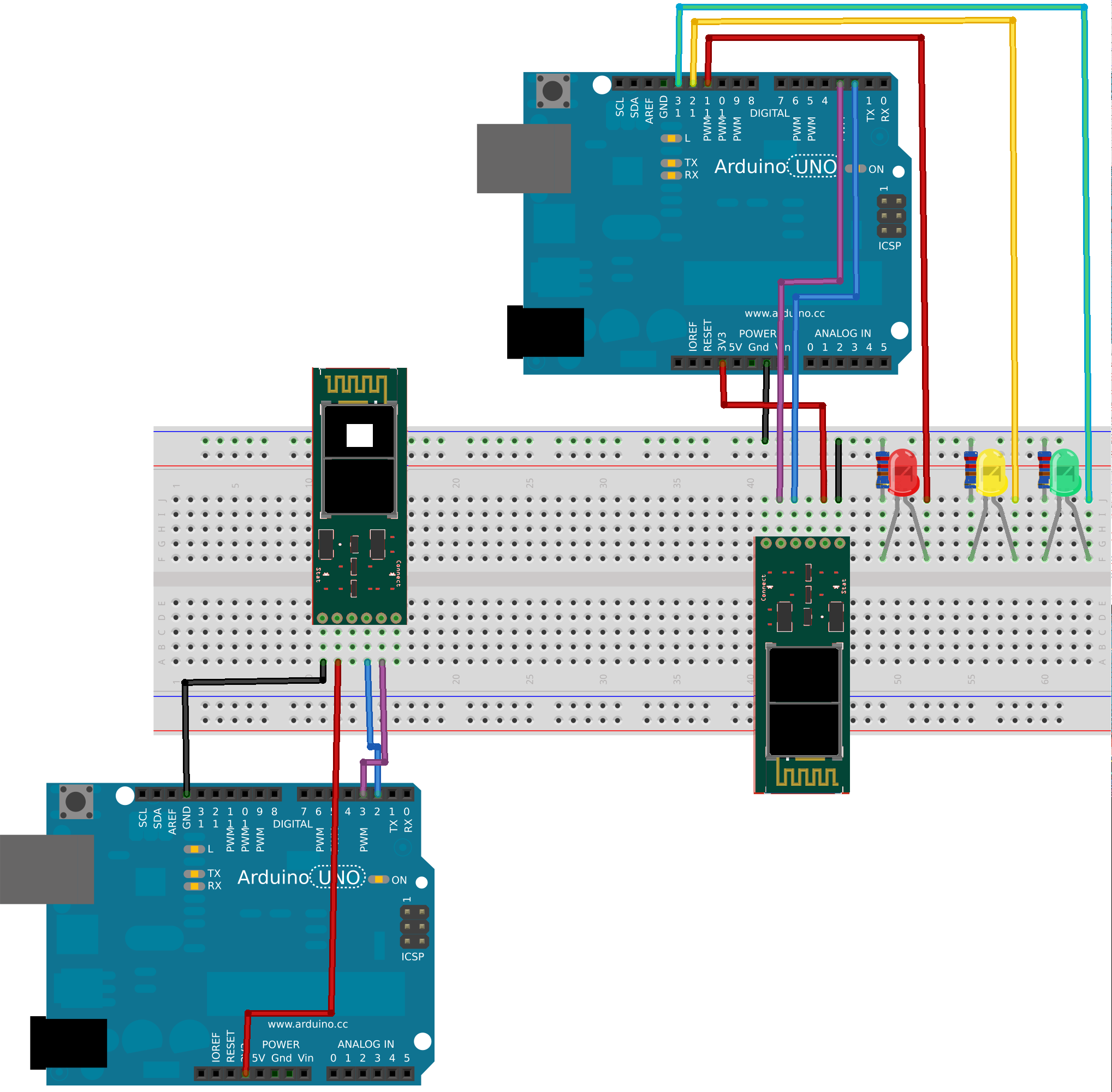
Es la opción más elegante y la que mayor alcance y tasa de transferencia de datos garantiza. Su uso es bastante sencillo, pues en última instancia nos provee de un puerto serie por donde enviaremos y recibiremos los datos intercambiados con otros módulos Zigbee. [3]

Para aquellos que se decidan por Zigbee, existe gran cantidad de documentación y ejemplos para las distintas plataformas de microcontroladores existentes en el mercado. Nuestro querido Arduino no podía ser menos y, como era de esperar, disponemos incluso de shields para poder usar módulos Zigbee en nuestros proyectos:

**[](http://3.bp.blogspot.com/-bhggzmUjo84/Us6w1cwJMjI/AAAAAAAABHk/Sy1Pk4nurJE/s1600/xbee_shield.jpg)**

**Cuadro 8: Shield de Arduino para un módulo XBee similar al visto anteriormente**

**Circuito**



**Cuadro9: circuito de conexión entre bluetooth**

Bluetooth es la norma que define un estándar global de comunicación inalámbrica que posibilita la transmisión de voz y datos entre diferentes equipos. La especificación de bluetooth define un canal de comunicación de máximo 720Kb/s con rango óptimo de 10 metros.

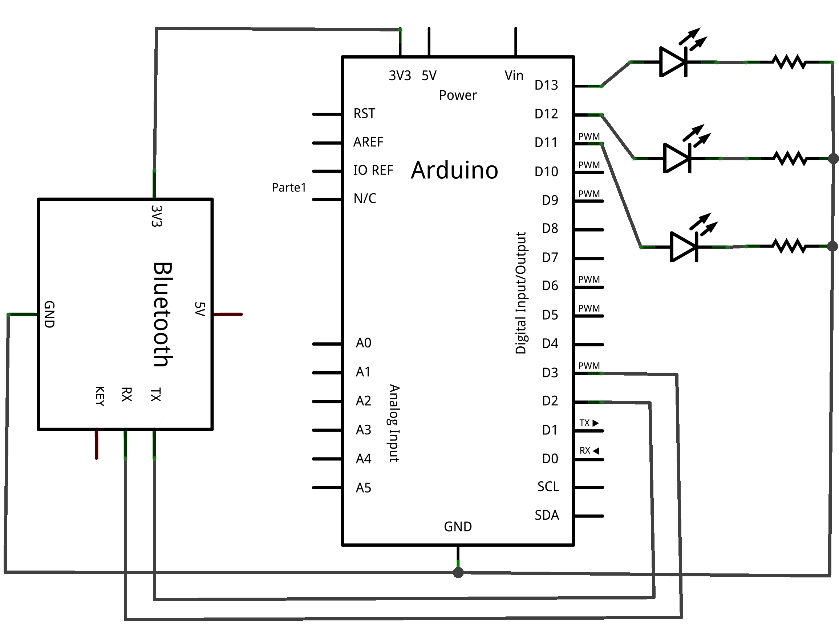
Tanto el programa como las conexiones para el caso del Módulo Serial Bluetooth Esclavo (SLAVE) son iguales a los tutoriales anteriores, por lo que se recomienda revisarlos antes:

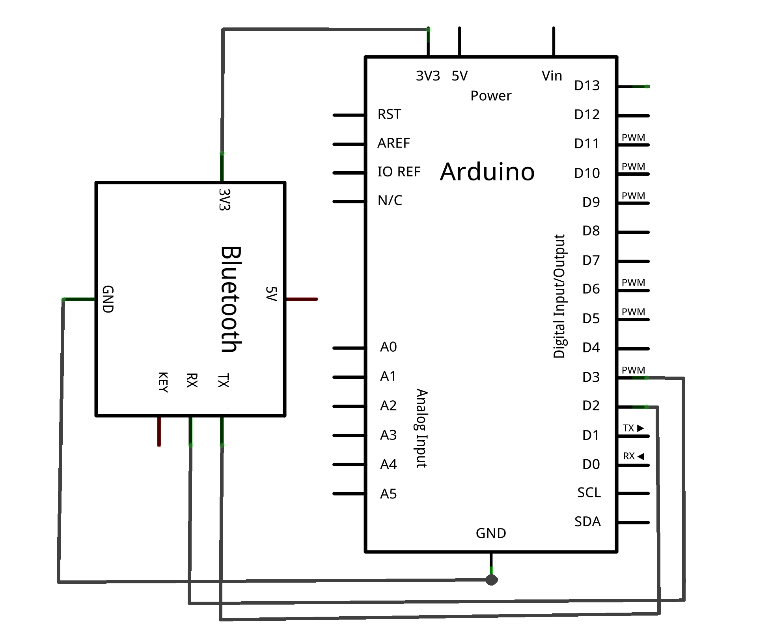
* [Comunicación Arduino-Android por Bluetooth](https://paruro.pe/aprende/comunicaci%C3%B3n-arduino-android-por-bluetooth)
* [Comunicación Arduino-PC mediante Bluetooth](https://paruro.pe/aprende/comunicaci%C3%B3n-arduino-pc-mediante-bluetooth)

En caso del Módulo Serial Bluetooth Maestro (HOST) sólo bastará con las conexiones primarias y ejecutar el programa.

**Esquemático**

Módulo Serial Bluetooth-SLAVE:





Módulo Serial Bluetooth-HOST

* 1. **Metodología y Resultados**

**Sensor de color TCS3200 con Arduino**

**1.- Para el sensor de color tcs3200 tendremos en cuenta algunas especificaciones como:**

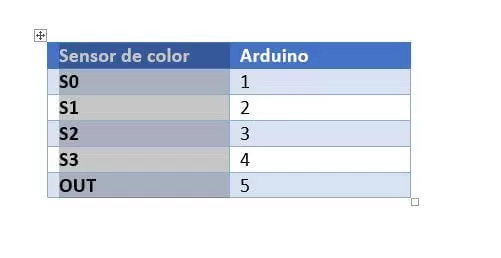
* Alta resolución de conversión de luz a frecuencia.
* Frecuencia de salida Programable en color y escala completa.
* Se comunica directamente con un microcontrolador.
* Voltaje de funcionamiento: 2.7-5.5 V.
* Rango de error típicamente de 0,2% a 50 kHz.
* Coeficiente de temperatura 200 ppm/°C.

**2.- Operamos con los materiales necesarios como:**

* [Tarjeta Arduino](https://hetpro-store.com/tarjetas-arduino/)
* [TCS3200 LC Tech](https://hetpro-store.com/sensor-de-color-rgb-tcs3200-lc-tech/)
* [Cables jumper macho-hembra](https://hetpro-store.com/cables-jumper-macho-hembra-40-unidades/)

**3.- Una vez obtenidos los materiales procedemos a investigar las conexiones que esta requiere con su respectivo puerto.**

La tabla siguiente muestra hacia como se deben hacer las conexiones del sensor hacia un Arduino UNO.

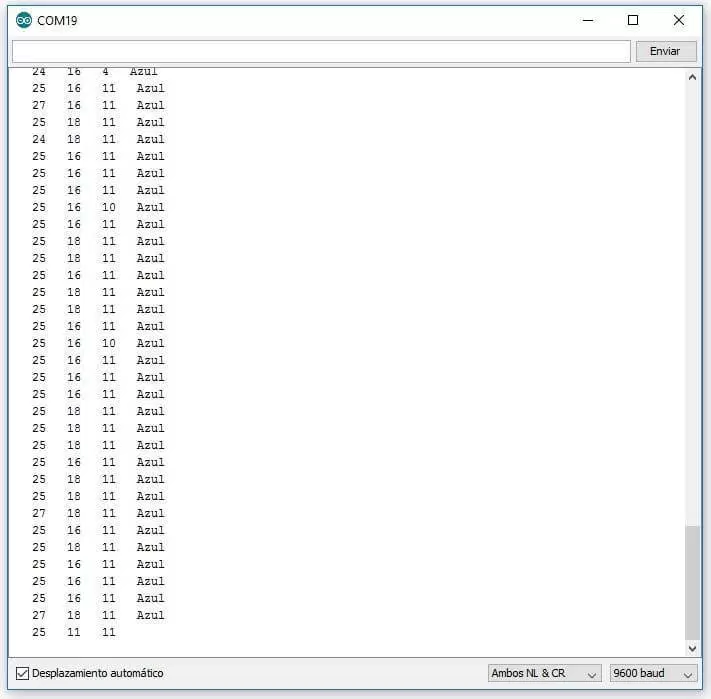


**Cuadro 10: tabla de conexiones del sensor de color con Arduino UNO**

### **PROGRAMACIÓN**

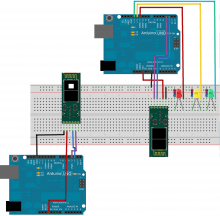
**4.- En la programación se abre el monitor serial y se muestra los valores de rojo, verde y azul, también se muestra cuando detecta el color rojo verde o azul, puedes modificar.**

|  |  |
| --- | --- |
|  | const int s0 = 1; |
|  | const int s1 = 2; |
|  | const int s2 = 3; |
|  | const int s3 = 4; |
|  | const int out = 5; |
|  | int rojo = 0; |
|  | int verde = 0; |
|  | int azul = 0; |
|  |  |
|  | void setup(){ |
|  | Serial.begin(9600); |
|  | pinMode(s0,OUTPUT); |
|  | pinMode(s1,OUTPUT); |
|  | pinMode(s2,OUTPUT); |
|  | pinMode(s3,OUTPUT); |
|  | pinMode(out,INPUT); |
|  | digitalWrite(s0,HIGH); |
|  | digitalWrite(s1,HIGH); |
|  | } |
|  |  |
|  |  |
|  | void loop(){ |
|  | color(); |
|  | Serial.print(" "); |
|  | Serial.print(rojo, DEC); |
|  | Serial.print(" "); |
|  | Serial.print(verde, DEC); |
|  | Serial.print(" "); |
|  | Serial.print(azul, DEC); |
|  |  |
|  | if (rojo < azul $$ verde > azul && rojo < 35) |
|  | { |
|  | Serial.println(" Rojo"); |
|  | } |
|  | else if (azul < rojo $$ azul < verde $$ verde < rojo) |
|  | { |
|  | Serial.println(" Azul"); |
|  | } |
|  |  |
|  | else if (rojo > verde $$ azul > verde ) |
|  | { |
|  | Serial.println(" Verde"); |
|  | } |
|  | else{ |
|  | Serial.println(" "); |
|  | } |
|  | delay(900); |
|  | } |
|  |  |
|  | void color() |
|  | { |
|  | digitalWrite(s2, LOW); |
|  | digitalWrite(s3, LOW); |
|  | rojo = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH); |
|  | digitalWrite(s3, HIGH); |
|  | azul = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH); |
|  | digitalWrite(s2, HIGH); |
|  | verde = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH); |
|  | } |



**Cuadro 11: Visualización del funcionamiento del sensor**

**5.- Para la comunicación entre el Arduino mediante Bluetooth realizamos el esquema luego de comprobar su funcionamiento.**



**Cuadro 12:** **Comunicación entre Arduino mediante Bluetooth**

**6.- Se debe saber que para dicha conexión se requiere de estos componentes:**

1 tarjetas Arduino Uno

1 dispositivo Serial Bluetooth- Maestro o Host

1 dispositivo Serial Bluetooth- Esclavo o Slave

3 Resistencia (220 ohm)

3 leds

7.- Posteriormente a la codificación de dicha conexión tanto del bluetooth esclavo como él maestro con el Arduino.

#include <Servo.h>

Servo servoMotor1;

Servo servoMotor2;

Servo servoMotor3;

const int s0 = 2;

const int s1 = 3;

const int s2 = 4;

const int s3 = 5;

const int out = 6;

int rojo = 0;

int verde = 0;

int azul = 0;

char dato=0;

void setup() {

pinMode(7,OUTPUT);

pinMode(8,OUTPUT);

pinMode(9,OUTPUT);

pinMode(10,OUTPUT);

Serial.begin(9600);

servoMotor1.attach(10);

//servoMotor1.write(0);

servoMotor2.attach(11);

//servoMotor2.write(0);

servoMotor3.attach(12);

//servoMotor3.write(0);

pinMode(s0,OUTPUT);

pinMode(s1,OUTPUT);

pinMode(s2,OUTPUT);

pinMode(s3,OUTPUT);

pinMode(out,INPUT);

digitalWrite(s0,HIGH);

digitalWrite(s1,HIGH);

}

int w;

void loop() {

if(Serial.available()>0){ //Verifica el primer caracter que ingresa al puerto serial

//estado = Serial.read();

dato = Serial.read();

}

color();

Serial.print(" ");

Serial.print(rojo, DEC);

Serial.print(" ");

Serial.print(verde, DEC);

Serial.print(" ");

Serial.print(azul, DEC);

if (rojo < azul && verde > azul && rojo < 35)

{

Serial.println(" Rojo");

w = 3;

}

else if (azul < rojo && azul < verde && verde < rojo)

{

Serial.println(" Azul");

w = 2;

}

else if (rojo > verde && azul > verde)

{

Serial.println(" Verde");

w = 1;

}

else{

Serial.println(" ");

}

delay(900);

servoMotor3.write(60);

//SELECCIÓN POR DEFECTO (AUTOMATICO) "A"

//SELECCIÓNA "B"

if(dato=='A'){

if(w == 1)//w=1 verde

{

delay(50);

servoMotor1.write(50);////

delay(3000);

servoMotor1.write(90);

//servoMotor3.write(60);

delay(100);

servoMotor1.write(90);

w=0;

}

if(w == 2) //w=2 azul

{

delay(200);

servoMotor2.write(50);

delay(3000);

servoMotor2.write(90);

delay(100);

servoMotor2.write(90);

w=0;

}

if(w == 3) //w=3 rojo

{

delay(300);

servoMotor1.write(90);

servoMotor2.write(90);

w=0;

}

}

if(dato=='B'){

if(dato == '1' && w == 1){

delay(100);

servoMotor1.write(0);

servoMotor2.write(0);

servoMotor3.write(60);

}

if(dato == '3' && w == 3)

{

servoMotor1.write(60);

servoMotor2.write(0);

servoMotor3.write(0);

}

if(dato == '5' && w == 2)

{

servoMotor1.write(0);

servoMotor2.write(60);

servoMotor3.write(0);

}

}

}

void color()

{

digitalWrite(s2, LOW);

digitalWrite(s3, LOW);

rojo = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);

digitalWrite(s3, HIGH);

azul = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);

digitalWrite(s2, HIGH);

verde = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);

}

* 1. **Recomendaciones**

Para obtener los resultados esperados se recomienda hacer el diseño de lo que se requiere hacer, buscando todo lo necesario para que el diseño funcione correctamente.

Es importante compilar bien la programación ya que es necesario calibrar tiempos t estados en cada instante para el desarrollo de los servos.

* 1. **Conclusiones**
* Se realizo una comunicación inalámbrica de manera exitosa.
* Investigamos el método de comunicación inalámbrica a utilizar mediante Arduino.
* Fue factible la capacidad al momento de implementar la comunicación para un buen funcionamiento.
* Indagamos en la ventaja de realizar las comunicaciones inalámbricas ya que no existen cables físicos: por lo tanto, no hay cables que se enreden, ni que entorpezcan la transpirabilidad o que molesten estéticamente.
  1. **Bibliografía**

# Bibliografía

|  |  |
| --- | --- |
| [1] | D. ROLDÁN MARTÍNEZ, COMUNICACIONES INALÁMBRICAS. UN ENFOQUE APLICADO., RA-MA EDITORIAL, 2004. |
| [2] | D. ROLDAN, COMUNICACIONES INALAMBRICAS, ESPAÑA: RA-MA, 2004. |
| [3] | J. DORDOIGNE, Redes informáticas: Nociones fundamentales, 4ta Edición, May 2014. |
| [4] | «Electrotenia,» 2010. [En línea]. Available: http://www.asifunciona.com/electrotecnia/ke\_resistencia/ke\_resistencia\_1.htm. |

**Anexos**

