

Abstract

El problema del diseño de la respuesta óptica de un material mediante la composición y nanoestructuración del mismo se puede plantear en términos de problema de como generar y combinar adecuadamente tres componentes principales distribuidos en el espectro visible. Por ejemplo, en la observación del color producido al iluminar una superficie, los tonos del color para el ojo humano estándar dependen principalmente de la reflectancia ponderada con la luz empleada para iluminar esa superficie y de las funciones triestímulos que caracterizan la sensibilidad de la retina del ojo humano a la luz observada. Un caso práctico para representar esta aproximación a la solución del problema es la generación de un color arbitrario en el mapa cromático estándar RGB (sRGB) empleando una combinación del leds (del inglés light emitting devices) RGB (del ingles red, green y blue). En este trabajo proponemos emplear una Raspberry Pi modelo B 3 [1] para lograr mediante pulsos modulados en el ancho de su período (PWM del ingles pulse width modulated) [2] un ajuste de color minimizando la diferencia en el mapa cromático (ΔE) entre un color propuesto y otro generado.

Definición del Problema

Controlar la alimentación de un led RGB para que su emisión sea de un color preestablecido con coordenadas en el mapa cromático standard.

Propuesta de Trabajo

- Selección de \vec{E}_1 con coordenada cromáticas (x, y) ,
- Obtención un espectro SPD que genere \vec{E}_1 .
- Medición del espectro I/I_0 del led RGB
- Control de la alimentación eléctrica p_R, p_G y p_B en cada terminal del led RGB (pulso PWM) [3]
- Mediante un código de optimización obtener p_R, p_G y p_B que haga mínima la diferencia entre el espectro del led RGB y el espectro del color \vec{E}_1 .
- Obtención de las coordenadas \vec{E}_2 del led RGB
- Medición de $\Delta E = |\vec{E}_2 - \vec{E}_1|$
- Configuración, cálculo y empleo de los valores óptimos de los pulsos PWM para comandar el led RGB desde una Raspberry Pi 3 [1]

Bibliotecas Computacionales

Codificación del programa con Python3 y utilización de bibliotecas:

- *GPIO* del inglés *General Purpose Input Output* [4] constituido en el modelo Pi 3 por 40 pines para la interacción de las raspberry pi con elementos pasivos, activos y microcontroladores externos en general.
- *colour-science*, conjunto de códigos escrito en lenguaje Python3 para implementar transformaciones de coordenadas cromáticas y espectros de reflectancia e irradiancia [5].
- *iminuit*. código para optimización, permite determinar parámetros de una función. Versión para lenguaje python [6]

Ingreso de Parámetros

1. Se establece el valor de las coordenadas (x, y) , en el color blanco estándar $D(65)$
2. Transformación $(x, y) \Rightarrow (X, Y, Z)$
3. Espectro de reflectancia de las coordenadas (X, Y, Z) vía el método Meng 2015 [7]
4. Los valores sRGB se definen por la transformación lineal, corregida por un factor no lineal caracterizada por γ a partir de coordenadas XYZ

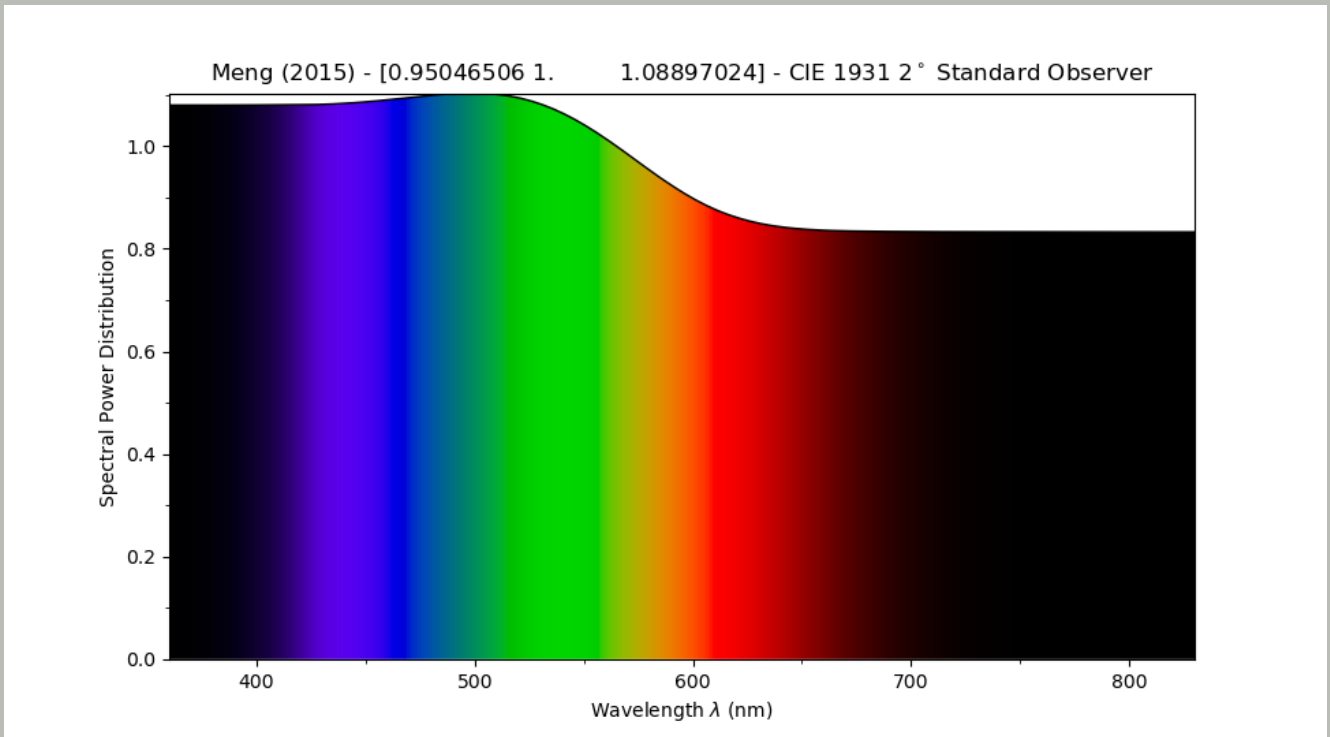


Figura: Espectro blanco ($x = 0,31, y = 0,33$) $\Rightarrow (X = 0,95, Y = 1, Z = 1,08)$

LEDS RGB

Los LEDS de tipo RGB permiten cambiar el color que emiten gracias a que disponen de tres *canales* (Rojo, Verde, Azul), para regular la intensidad a cada color y uno correspondiente al polo positivo de alimentación.

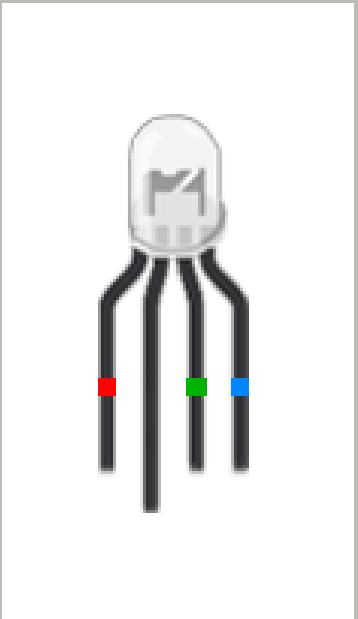
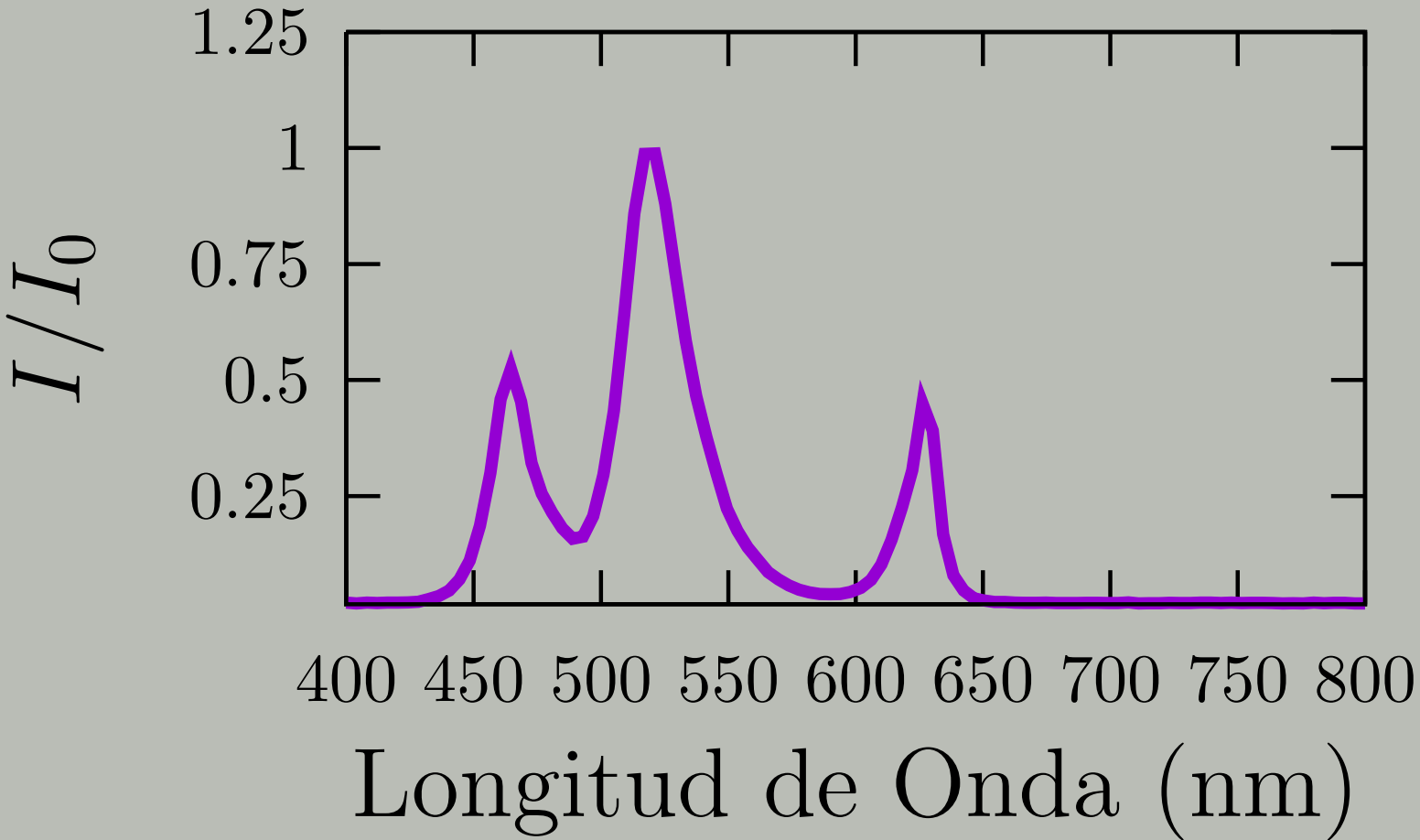


Figura: Izquierda: Espectro de irradiancia I del led RGB normalizado al valor máximo I_0 obtenido con el espectrómetro OceanOptics USB4000. Derecha: Esquema del led donde se indica con cada color los tres canales RGB y el pin mas largo es el ánodo de alimentación de 3.3v

Para modificar la intensidad del brillo de un canal usaremos el PWM, esto nos permite variar la longitud de los pulsos eléctricos manteniendo el número de pulsos por segundo.

Codificación Gpio

1. Se declaran los tres pines del leds RGB:
2. Seleccionando su codificación `BOARD` (alternativa `BCM`)
3. Estos deben ser del tipo PWM \Leftarrow 12, 32 y 33 Raspberry Pi 3 modelo B
4. Se establecen como pines de salidas, a una frecuencia de 50 Hz:

Función de Ajuste f

- Modelamos I/I_0 con los parámetros p_R, p_G y p_B pesos de tres gaussianas centradas en 460, 520 y 630 nm $\Rightarrow \tilde{I} = I(p_R, p_G, p_B)$
- $f \Leftarrow$ error cuadrático promedio entre \tilde{I} y SPD
- Uso de una estructura de datos que permite definir en forma dinámica la función de ajuste f

Optimización

- Funciones de la librería *iminuit*
- Inicialización de los parámetros p_R, p_G y p_B
- Uso de la función *migrad()* de la librería *iminuit* para obtener el reporte de optimización

PWM

- Se calcula el porcentaje individual de los parámetros p_R, p_G y p_B
- Se configura la señal PWM con los porcentajes correspondientes de cada color, a través de la función *ChangeDutyCycle()* de la librería *GPIO* se establece la relación de encendido, donde 0 significa que el LED estará completamente encendido durante totalidad del ciclo, y con 100 se establece como apagado

Coordenadas Cromáticas Ajuste

- A partir de la coordenada preestablecida $\vec{E}_1 = (0,31, 0,33)$ se obtiene la coordenada $\vec{E}_2 = (0,28, 0,28)$ a través de la función de ajuste
- Las coordenadas se representan gráficamente en el diagrama de cromaticidad CIE (Commission internationale de l'éclairage) estándar 1931

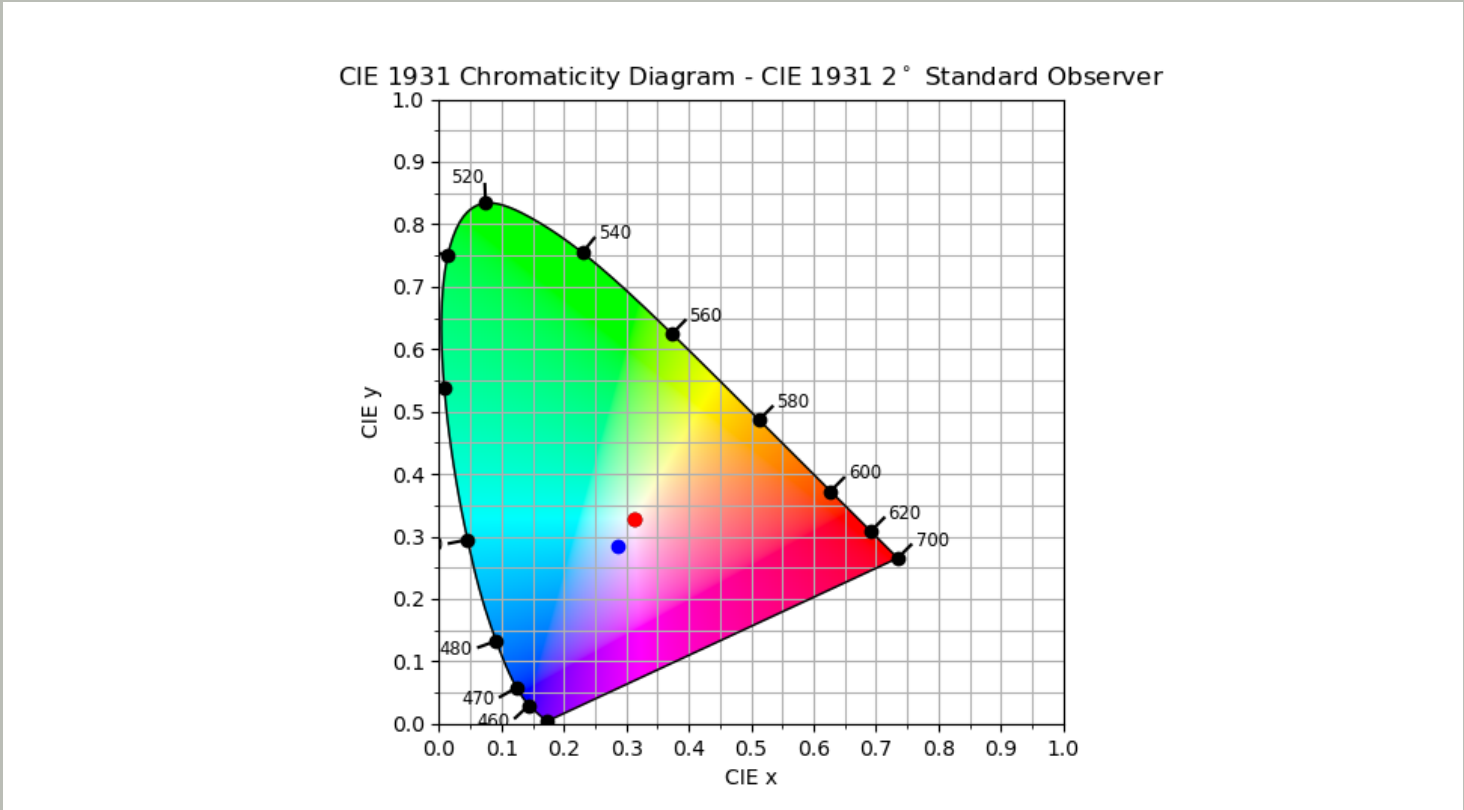


Figura: Se comparan en el mapa de coordenadas cromáticas el ajuste entre el punto rojo que indica el dato ingresado \vec{E}_1 y el punto azul el valor de \vec{E}_2 generado

Representación en led RGB

- Se usan los valores de los pulsos PWM para emitir el color ajustado \vec{E}_2 a través del led RGB

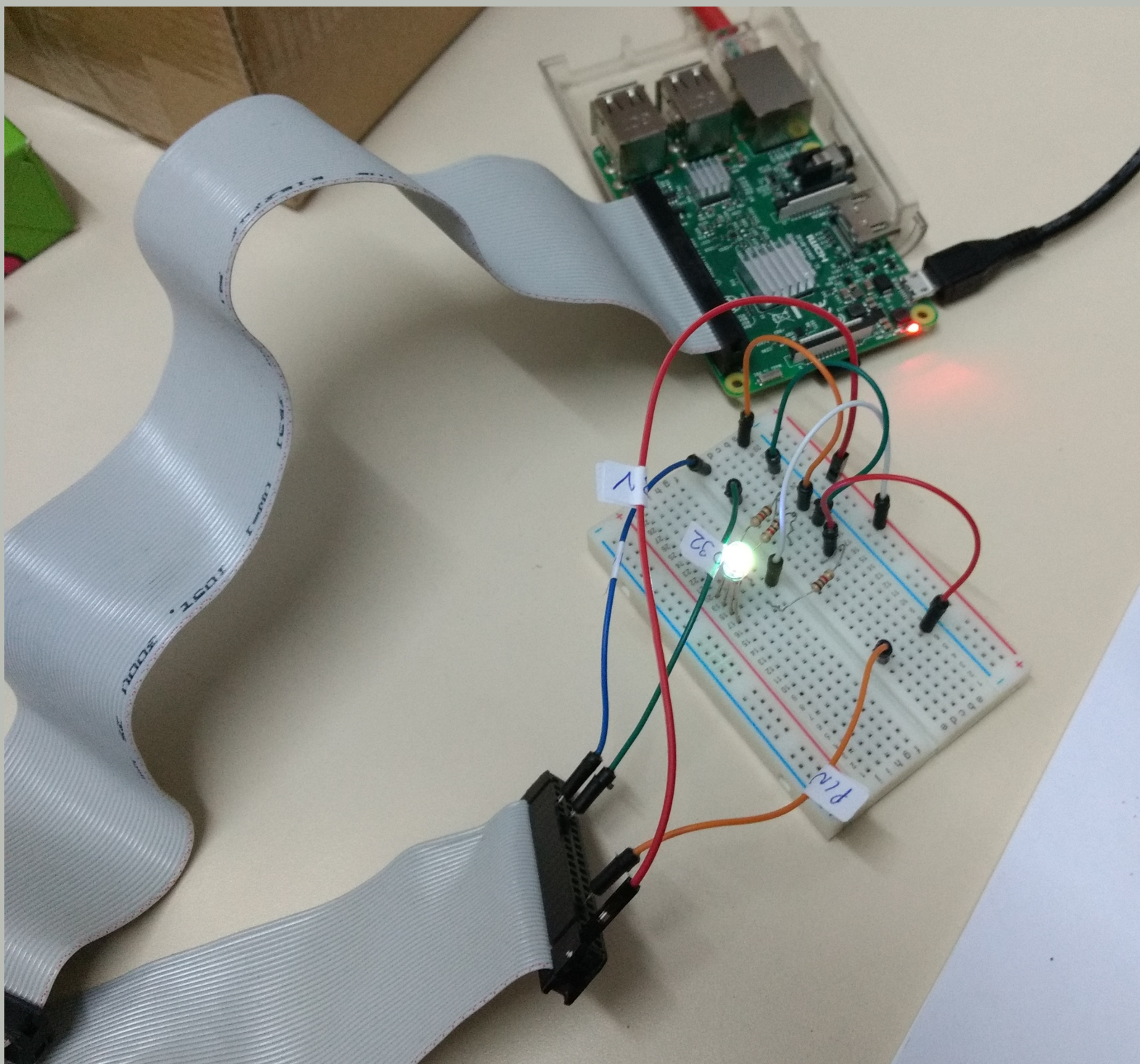


Figura: Se observa el encendido del led RGB al color \vec{E}_2 y las conexiones a la raspberry pi, por medio de un cable IDE(Integrated Drive Electronics) y el modulo zócalos de conexión con resistencias de un 1 kohms a cada canal de color

Conclusión

- Se considera una diferencia aceptable $\Delta E \leq 0,5$
- En nuestro caso planteamos reproducir un color blanco estándar (D 65)
- Empleamos una Raspberry Pi 3 modelo B para comandar el led RGB mediante 3 pulsos de ancho modulado PWM
- Obtenemos que el led RGB reproduce ese blanco con un $\Delta E \approx 0,5$

Referencias

[1] <https://www.raspberrypi.org/>.

[2] <http://digital.ni.com/public.nsf/allkb/AA1BDEA4AA224E3E86257CE400707527>.

[3] <https://barrgroup.com/Embedded-Systems/How-To/PWM-Pulse-Width-Modulation>.

[4] <http://www.mosaic-industries.com/embedded-systems/microcontroller-projects/raspberry-pi/gpio-pin-electrical-specifications>.

[5] Thomas Mansencal, Michael Mauderer, Michael Parsons, Luke Canavan, Sean Cooper, Nick Shaw, Kevin Wheatley, Katherine Crowson, and Ofek Lev. Colour 0.3.11, February 2018.

[6] <http://iminuit.readthedocs.io/en/latest/api.html>.

[7] Johannes Meng, Florian Simon, Johannes Hanika, Carsten DachsbacherB. Physically Meaningful Rendering using Tristimulus Colours. Eurographics Symposium on Rendering , Volume 34 (2015), Number 4. doi:10.1111/cgf.12676.