

Escuela Superior Politécnica del Litoral

Facultad de Ingeniería en Electricidad y Computación

Diseño e implementación de un sistema de comunicación OCC basado en dispositivos
inteligentes con fines pedagógicos

Proyecto Integrador

Previo la obtención del Título de:

Ingeniero en Telecomunicaciones

Presentado por:

Lisbeth Abigail Ramirez Bermeo

Jorge Andrés González Narváez

Guayaquil - Ecuador

Año: 2023

Agradecimientos

Queremos expresar nuestro más sincero agradecimiento a la Escuela Superior Politécnica del Litoral por brindarnos la oportunidad de realizar este proyecto integrador. También queremos extender nuestra gratitud a la PhD. Patricia Chávez por guiarnos a lo largo de este arduo camino lleno de dificultades. Agradecemos especialmente a nuestros padres por el apoyo y sacrificio constante, lo cual ha sido base nuestro éxito.

Dedicatoria

Este proyecto está dedicado a Dios por iluminar cada paso de este viaje de aprendizaje y crecimiento. A mis padres, Jaqueline Bermeo y Wilson Ramírez, por brindarme su esfuerzo, dedicación y amor incondicional. A mi hermana, Rebeca Ramírez, quien siempre me demostró su apoyo incondicional en las decisiones más importantes de mi vida. A José Daniel Plúas Segura, quien ha dejado una huella imborrable en mi vida y ha hecho posible este logro. A Magali Ramirez, por haber sido mi segunda madre y dedicar muchos años de su vida a cuidar de mí. A mis amigos por acompañarme en esta travesía, llenando mis días de risas y recuerdos inolvidables. Por último, a Roberto, cuyos recuerdos aún viven en mi corazón a pesar de ya no tenerte a mi lado.

Lisbeth Abigail Ramirez Bermeo

Dedicatoria

En el camino de esta travesía académica, me siento profundamente agradecido por tener a mi lado a seres queridos cuyo apoyo ha sido un faro constante, guiándome en cada paso y compartiendo mis triunfos y desafíos. A mis amados padres, cuyo amor incondicional y sacrificio han sido la base sobre la cual he construido mis logros. A mis abuelos, cuyo cariño y sabiduría han sido un regalo constante en mi vida. A mi hermana, cuyas lecciones y orientación han sido valiosas brújulas en mis momentos de incertidumbre. A mi querida sobrina, cuyo cariño puro y genuino ha sido una fuente inagotable de inspiración. Y a todos mis amigos y compañeros que han sido parte inseparable de mi travesía universitaria

Jorge Andrés González Narváez

Declaración expresa

"Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; Lisbeth Abigail Ramirez Bermeo, Jorge Andrés González Narváez damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual".



Lisbeth Abigail Ramirez
Bermeo



Jorge Andrés González
Narváez

EVALUADORES

.....
Verónica Soto MSc.

Profesor de la materia

.....
Patricia Chavez Ph.D.

Profesor tutor

RESUMEN

El siguiente proyecto de titulación se centra en el desarrollo de un sistema de comunicación OCC con propósitos pedagógicos. El objetivo principal de este proyecto es facilitar la comprensión de las tecnologías de comunicación mediante luz visible para estudiantes novatos, a través de un enfoque didáctico y fácil de comprender.

El sistema de comunicación se compone de un transmisor que utiliza modulación por PWM. En este transmisor, se emplea un microcontrolador ESP32 configurado mediante Arduino para modular los datos proporcionados por el usuario. La modulación se basa en modificar el ciclo de trabajo de una tira LED RGB, lo que permite la transmisión efectiva de información dado al uso exclusivo de dos canales de comunicación y, gracias a la IDE de Arduino, el usuario puede ingresar los datos que desean transmitirse, los cuales son luego modulados en la tira LED RGB. El receptor óptico consiste en la cámara de un teléfono celular. Esta cámara está conectada a una computadora mediante la aplicación iVCam y a través de Visual Studio Code en conjunto con la programación en Python, se procesan y representan los datos captados por el dispositivo.

En términos de las configuraciones aplicadas, se han obtenido resultados positivos a una distancia de 20 cm entre el transmisor y el receptor. Gracias a la elección de los canales rojo y azul como medios de transmisión, se ha logrado una recepción efectiva de los datos transmitidos. La señal transmitida opera a una frecuencia de 240 Hz, lo que permite la comunicación de este sistema siendo imperceptible para el ojo humano.

En resumen, este dispositivo de comunicación OCC ha permitido una integración efectiva en la operación de sistemas inalámbricos de este tipo. Asimismo, esta propuesta facilitará la comprensión de los estudiantes acerca de estos sistemas, alentándolos a explorar e investigar estas innovadoras tecnologías de vanguardia.

ABSTRACT

The following degree project focuses on the development of an OCC communication system for pedagogical purposes. The main objective of this project is to facilitate the understanding of visible light communication technologies for novice students, through a didactic and easy-to-understand approach.

The communication system consists of a transmitter that uses PWM modulation. In this transmitter, an Arduino-configured ESP32 microcontroller is used to modulate the data provided by the user. The modulation is based on modifying the duty cycle of an RGB LED strip, which allows the effective transmission of information given the exclusive use of two communication channels and, thanks to the Arduino IDE, the user can enter the data they want transmitted, which are then modulated in the RGB LED strip. The optical receiver consists of a cell phone camera. This camera is connected to a computer through the iVCam application and through Visual Studio Code in conjunction with Python programming, the data captured by the device is processed and represented.

In terms of the configurations applied, positive results have been obtained at 20 cm between the transmitter and the receiver. Thanks to the choice of the red and blue channels as transmission media, an effective reception of the transmitted data has been achieved. The transmitted signal operates at a frequency of 240 Hz, which allows the communication of this system to be imperceptible to the human eye.

In short, this OCC communication device has enabled effective integration into the operation of wireless systems of this type. Likewise, this proposal will facilitate the understanding of students about these systems, encouraging them to explore and investigate these innovative cutting-edge technologies.

ÍNDICE GENERAL

EVALUADORES	6
RESUMEN	I
ABSTRACT	II
ABREVIATURAS	V
SIMBOLOGÍA.....	V
ÍNDICE de figuras	VI
ÍNDICE de TABLAS	1
ÍNDICE de Anexos	1
Capítulo 1.....	2
1.1 Introducción	3
1.2 Descripción del problema	4
1.3 Justificación del problema	4
1.4 Objetivos	5
1.4.1 Objetivo general.....	5
1.4.2 Objetivos específicos	5
1.5 Marco teórico	5
1.5.1 OWC.....	5
1.5.2 Visible Light Communication (VLC).....	5
1.5.3 Optical Camera Communication (OCC)	6
1.5.4 Modulación por ancho de pulso (PWM)	7
1.5.5 OpenCV.....	7
Capítulo 2.....	9
2 Metodología.....	10
2.1 Diagrama conceptual de la solución.....	10
2.1.1 Transmisión.....	11
2.1.2 Recepción.....	11

2.2	Escenarios propuestos.....	11
2.2.1	Solución 1: Transmisión con modulación OOK, codificación Manchester y LEDs blancos.	11
2.2.2	Solución 2: Transmisión con modulación PWM y LEDs RGB.....	13
2.3	Diagrama de flujo de la solución propuesta.	15
Capítulo 3	18
3	Resultados y análisis.....	19
3.1	Implementación de la solución propuesta.....	19
3.1.1	Transmisor	19
3.1.2	Receptor	20
Capítulo 4	35
4	Conclusiones y recomendaciones	36
4.1	Conclusiones.....	36
4.2	Recomendaciones.....	37
4.3	Trabajos Futuros	37
Anexos	38
Referencias	56
APÉNDICE	58

ABREVIATURAS

OCC Optical Camera Communication
VLC Visible Light Communications
OOK On-Off-Keying
LED Light Emitting Diode
OWC Optical Wireless Communication
PWM Pulse Width Modulation
OpenCV Open Source Computer Vision Library
RGB Red, Green, Blue
SMD Surface Mount Device
PCB Printed Circuit Board

SIMBOLOGÍA

us microsegundos
V Voltios
Hz Hertz

ÍNDICE DE FIGURAS

Figura 1 Sistema OCC	10
Figura 2 Sistema OCC con transmisión por LEDs blancos.....	12
Figura 3 Procesamiento de la información	12
Figura 4 Procesamiento de la información.....	14
Figura 5 Sistema OCC con transmisión por LEDs RGB.....	14
Figura 6 Diagrama de flujo del transmisor.....	16
Figura 7 Diagrama de flujo del receptor	17
Figura 8 Prototipo inicial del sistema OCC.	21
Figura 9 Señal transmitida en el canal rojo.....	22
Figura 10 Señal transmitida en el canal azul.	22
Figura 11 Lámpara vista desde la cámara web.	23
Figura 12 Aplicación móvil desarrollada en base a ImagePro	24
Figura 13 Ajustes de cámara.....	25
Figura 14 Transmisión del “0” visto desde la cámara del teléfono.	26
Figura 15 Imagen binaria.	27
Figura 16 Imagen binaria recortada.....	27
Figura 17 Decodificación del “0”.....	28
Figura 18 Visualización del número decodificado en el teléfono celular.....	29
Figura 19 Esquemático modificado	30
Figura 20 Diagrama esquemático final	30
Figura 21 Diseño de caja	31
Figura 22 Sistema OCC final.....	32
Figura 23 Señal PWM canal rojo.....	33
Figura 24 Transmisión del “1” visto desde la cámara del teléfono.	38
Figura 25 Transmisión del “2” visto desde la cámara del teléfono.	38
Figura 26 Transmisión del “3” visto desde la cámara del teléfono.	39
Figura 27 Transmisión del “4” visto desde la cámara del teléfono.	39
Figura 28 Transmisión del “5” visto desde la cámara del teléfono.	40
Figura 29 Transmisión del “6” visto desde la cámara del teléfono.	40
Figura 30 Transmisión del “7” visto desde la cámara del teléfono.	41
Figura 31 Transmisión del “8” visto desde la cámara del teléfono.	41

Figura 32 <i>Transmisión del “9” visto desde la cámara del teléfono.</i>	42
Figura 33 <i>Transmisión del “10” visto desde la cámara del teléfono.</i>	42
Figura 34 <i>Transmisión del “11” visto desde la cámara del teléfono.</i>	43
Figura 35 <i>Transmisión del “12” visto desde la cámara del teléfono.</i>	43
Figura 36 <i>Transmisión del “13” visto desde la cámara del teléfono.</i>	44
Figura 37 <i>Transmisión del “14” visto desde la cámara del teléfono.</i>	44
Figura 38 <i>Transmisión del “15” visto desde la cámara del teléfono.</i>	45
Figura 39 <i>Decodificación del “1”</i>	45
Figura 40 <i>Decodificación del “2”</i>	46
Figura 41 <i>Decodificación del “3”</i>	46
Figura 42 <i>Decodificación del “4”</i>	47
Figura 43 <i>Decodificación del “5”</i>	47
Figura 44 <i>Decodificación del “6”</i>	48
Figura 45 <i>Decodificación del “7”</i>	48
Figura 46 <i>Decodificación del “8”</i>	49
Figura 47 <i>Decodificación del “9”</i>	49
Figura 48 <i>Decodificación del “10”</i>	50
Figura 49 <i>Decodificación del “11”</i>	50
Figura 50 <i>Decodificación del “12”</i>	51
Figura 51 <i>Decodificación del “13”</i>	51
Figura 52 <i>Decodificación del “14”</i>	52
Figura 53 <i>Decodificación del “15”</i>	52

ÍNDICE DE TABLAS

Tabla 1 <i>Números enviados</i>	53
Tabla 2 <i>Tabla de costos de implementación del sistema OCC</i>	34

ÍNDICE DE ANEXOS

Anexo A. Transmisión de datos. imágenes de los números 24 al 38.....	38
Anexo B. Decodificación de datos. Imágenes de los números 39 al 53.....	45
Anexo C. Datos enviados en las pruebas de transmisión y recepción.....	53

CAPÍTULO 1

1.1 Introducción

Los sistemas de Comunicación Óptica por Cámara (OCC) utilizan receptores fotosensibles ante la luz visible; a diferencia de las otras tecnologías que se encuentran dentro de OWC, esta proporciona una mejor transmisión de datos, permite el acceso simultáneo de varios usuarios a la información transmitida y comunicación en el dominio del color. Las limitaciones no son una excepción en esta tecnología, entre ellas está el corto alcance que posee y la sensibilidad a la iluminación. (N. Saeed, S. Guo, K. H. Park, T.Y. Al-Naffouri, M.S. Alouini, 2019)

La instrucción de los conceptos relacionados con la comunicación óptica a través del uso de cámaras OCC basa su importancia en la amplitud de sus implicaciones y al potencial que posee para remodelar el panorama de dicho campo de estudio. Este adelanto tecnológico está encaminado a alterar drásticamente la forma en que se comunica y se transmite información, permitiendo percibir los beneficios que la luz ostenta como portadora primordial de datos en diversas aplicaciones. (M. Z. Chowdhury, M. T. Hossan, A. Islam and Y. M. Jang, 2018)

Al presentar a los estudiantes OCC, los educadores los exponen a un campo en constante evolución que posee la capacidad de cambiar nuestra interpretación de los sistemas de comunicación. El énfasis pedagógico puesto en OCC brinda a los estudiantes oportunidades para adquirir conocimientos especializados en un área de creciente importancia. OCC confiere una multitud de ventajas, incluida la capacidad de transmisión de datos aumentada, el acceso simultáneo para múltiples usuarios y la comunicación facilitada dentro del ámbito cromático.

En el presente contexto, se enfatiza la implementación de sistemas de comunicación óptica por cámara, dotados de la habilidad de desempeñar una doble función: la transmisión de datos y la iluminación concomitante. Esta convergencia entre la comunicación y la iluminación se consolida como una estrategia innovadora y eficaz, que posibilita la ejecución de transmisiones sin perturbar ni menoscabar las funcionalidades preexistentes.

La implementación de sistemas de comunicación óptica por cámara radica en el empleo de diodos emisores de luz (LEDs) con capacidad de generar una amplia gama cromática

gracias a su configuración RGB (rojo-verde-azul). Este enfoque no solo permite obtener iluminación blanca y amarilla, sino que también salvaguarda la posibilidad de mantener tres canales de comunicación simultáneos, engendrando una ventaja estratégica en términos de versatilidad y adaptabilidad.(CristoJV, 2020).

1.2 Descripción del problema

El esfuerzo de integrar los sistemas OCC dentro del panorama educativo conlleva ciertos desafíos, que obstaculizan la capacidad de los estudiantes para entender la gran extensión de beneficios que ofrece esta tecnología de punta. A pesar del progreso social, donde los avances tecnológicos prevalecen, la implementación práctica de los sistemas OCC ha quedado atrás de sus contrapartes de transmisión inalámbrica. Este inconveniente se deriva de una confluencia de factores, que abarca limitaciones de infraestructura formidables, una capacitación y familiaridad insuficientes con el tema, inquietudes relacionadas con la seguridad y la privacidad que implica la implementación de un sistema comunicación inalámbrico.

Dentro del ámbito de los propósitos educativos y el entorno del aprendizaje remoto, la búsqueda de la adopción de OCC se enfrenta a problemas que ofuscan su integración. El más importante de estos obstáculos es la integración de los exigentes requisitos de OCC con las limitaciones de infraestructura existentes.

1.3 Justificación del problema

La evolución tecnológica que se ha presentado en el transcurso de los años ha sido un verdadero desafío, por este motivo es necesario que los estudiantes sean preparados de manera correcta para enfrentarse a estos cambios. La implementación de los sistemas OCC en el ámbito educativo promueve la alfabetización digital, impulsa la creatividad, estimula la innovación en el estudiante y le permite desarrollar autonomía. (Izquierdo, A. M., 2021) Además, con la continua evolución digital, la comprensión de la manera en que funcionan los sistemas de Comunicaciones Ópticas por Luz permite al estudiante adaptarse a futuro al entorno laboral y poder aprovechar las oportunidades que este les brinda. (Los efectos de la tecnología en el trabajo, 2023)

1.4 Objetivos

1.4.1 Objetivo general

Desarrollar un algoritmo que permita detectar y analizar la transmisión de datos empleando OCC.

1.4.2 Objetivos específicos

1. Investigar sobre los métodos de modulación y codificación para la transmisión y recepción de datos mediante OCC.
2. Diseñar e implementar un sistema OCC (Optical Camera Communication) que permita la transmisión y recepción de datos tipo texto a 20 cm de distancia.
3. Desarrollar un algoritmo en Python que permita la detección y visualización de caracteres utilizando el teléfono como fotorreceptor.
4. Realizar un análisis del procedimiento de uso y manipulación del sistema OCC con la finalidad de mejorar su eficacia.

1.5 Marco teórico

1.5.1 OWC

OWC que en español significa comunicación óptica por ondas, surge como un tipo de tecnología de transmisión de datos utilizando el aire como su medio de propagación. Esta tecnología abarca un grupo de tecnologías, pero las que poseen una mayor relevancia son: VLC o Comunicación por Luz Visible y OCC o Comunicación Óptica por Cámara la cual es un tipo de tecnología específico de VLC. (M. Z. Chowdhury, M. T. Hossan, A. Islam and Y. M. Jang, 2018).

1.5.2 Visible Light Communication (VLC)

La comunicación por luz visible (VLC) se está convirtiendo en una alternativa para la tecnología inalámbrica de próxima generación por ofreciendo bajo costo, ancho de banda no regulado y ubicuo soporte de infraestructuras. Esta tecnología está prevista para ser utilizada en una amplia gama de aplicaciones tanto en interiores como en exteriores. VLC utiliza diodos emisores de luz (LED) (Rajan Sagotra, 2013). La ventaja primordial a estos sistemas radica en su utilización de infraestructuras de iluminación como medio de transmisión de datos. Esta innovadora tecnología amalgama hábilmente la capacidad lumínica con la capacidad de

transmitir información, lo cual posibilita un aprovechamiento óptimo de los recursos disponibles. Asimismo, se traduce en una robusta salvaguardia de la información y una aceleración sustancial en la velocidad de transmisión. (C. Q. Sánchez, 2013). Al hacer uso de las instalaciones de luces LED previamente creadas se puede implementar una red de comunicación sin la necesidad de armar una infraestructura adicional. La luz visible es la forma en que la radiación electromagnética con longitudes de onda en un rango particular es interpretada por el cerebro humano. El espectro de luz visible cubre longitudes de onda de 380 nm a 750 nm.

1.5.3 Optical Camera Communication (OCC)

OCC se deriva de manera directa de VLC, ya que al igual que este último, se aprovecha la luz visible para la transmisión de datos utilizando el aire como el medio por el cual se envía la información. OCC se diferencia de otros tipos de tecnologías de transmisión inalámbrica en su receptor, ya que utiliza la cámara de un dispositivo electrónico para demodular y decodificar las señales emitidas por el transmisor. La recepción simultánea de información es una de las ventajas de este tipo de sistemas, ya que varios usuarios pueden visualizar la información transmitida con la ayuda de aplicaciones adecuadas para el dispositivo que se desea utilizar.

La definición de los estados que se pueden presentar en los LEDs ayuda a mejorar la velocidad de transmisión de datos, en el caso de los LEDs RGB, se tiene la posibilidad de asignar un mayor número de estados, e incluso definir varios canales de transmisión, asignando una frecuencia en la cual el encendido y apagado de estos sea casi imperceptible; en estos sistemas también se puede implementar más de 1 lámpara para mejorar dicha transmisión. (CristoJV,2020)

1.5.4 Modulación por ancho de pulso (PWM)

Modulación de ancho de pulso (PWM) o modulación de duración de pulso (PDM) o modulación de tiempo de pulso (PTM) es un esquema de modulación analógica en el que la duración o el ancho o el tiempo de la portadora del pulso varía proporcionalmente a la amplitud instantánea de la señal del mensaje. El ancho del pulso varía en este método, pero la amplitud de la señal permanece constante. Los limitadores de amplitud se utilizan para hacer constante la amplitud de la señal. Estos circuitos recortan la amplitud, a un nivel deseado y, por lo tanto, el ruido es limitado. (Francesco Vasca, 2012)

En la comunicación por luz visible, PWM se basa en estados de encendido y apagado del LED por el cual se está transmitiendo; esta modulación permite manipular el porcentaje del ciclo de trabajo o conocido también como Duty Cycle. Por lo general, PWM utiliza un valor del 50% de encendido, y el restante en apagado. (Fluke, 2016) Esta fluctuación en los ciclos de trabajo permite mejorar la transmisión de la información de manera binaria. Los pulsos de luz con la variación en su ancho permiten representar diferentes bits de datos, también se pueden asignar estos datos a los diferentes estados que se definan en un LED. (Francesco Vasca, 2012)

1.5.5 OpenCV

OpenCV, que por sus siglas significa “Open Source Computer Vision Library”, es una biblioteca de programación ampliamente utilizada en el campo de procesamiento digital de imágenes. Esta librería fue desarrollada por Intel en el año 2000 y actualmente se mantiene como Código abierto para el uso libre de usuarios de la comunidad.

La librería OpenCV ofrece un entorno avanzado para la creación de aplicaciones de visión por computadora en tiempo real. Proporciona herramientas para manejar estructuras de datos, procesar y examinar imágenes, realizar análisis estructurales, y más. Esta herramienta simplifica significativamente la comprensión y aplicación de diversas técnicas de visión por computadora,

tanto en entornos educativos como de investigación, al abstraer al programador de las particularidades de los diversos sistemas de visión disponibles. (V. M. Arévalo, 2002)

CAPÍTULO 2

2 METODOLOGÍA

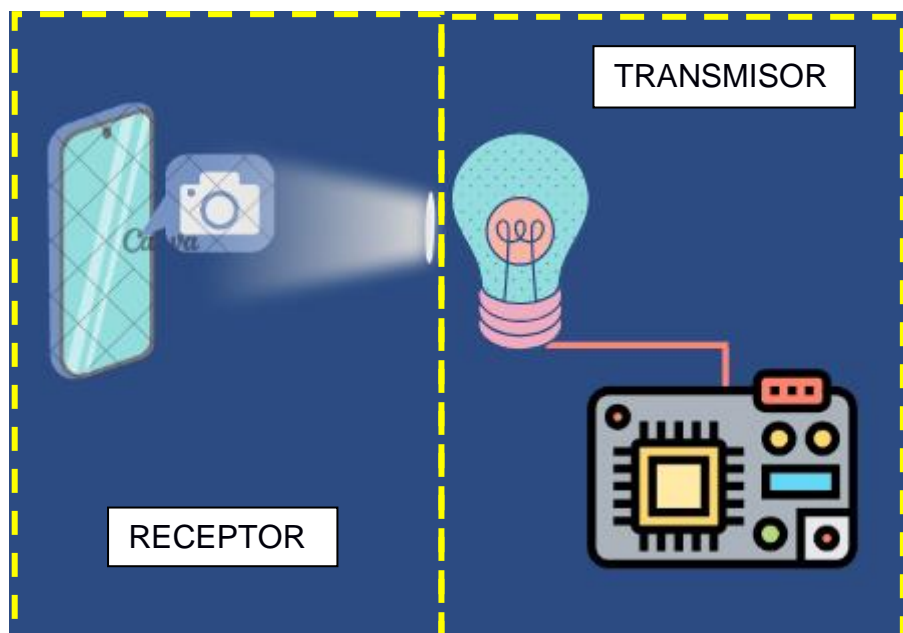
Este capítulo detalla las posibles soluciones presentadas al cliente para llevar a cabo la implementación del sistema OCC; cada una presenta diferentes técnicas, componentes y procedimientos para entablar la comunicación requerida.

2.1 Diagrama conceptual de la solución.

En la Figura 1 se puede visualizar de una manera más detallada y organizada el prototipo implementado para la transmisión y recepción de datos, siendo el foco una referencia a los LEDs RGB utilizados en el sistema.

Figura 1

Sistema OCC



Nota. En la figura se muestra un sistema de transmisión y recepción de datos con luz visible y un dispositivo móvil respectivamente.

2.1.1 Transmisión.

El transmisor está conformado por un microcontrolador ESP32 y LEDs SMD, dentro de la placa programable la información a transmitir es codificada de modo que se envíen bits y varíe la intensidad del LED dependiendo del carácter enviado.

2.1.2 Recepción.

El receptor del sistema implementado es el teléfono celular, la cámara de este recibe las señales que emiten los LEDs RGB, pero para poder decodificar dichas señales es necesario un algoritmo en Python, que contenga la librería OpenCV, el cual permite llevar a cabo un óptimo procesamiento de las imágenes capturadas y así mostrar en pantalla los datos recibidos.

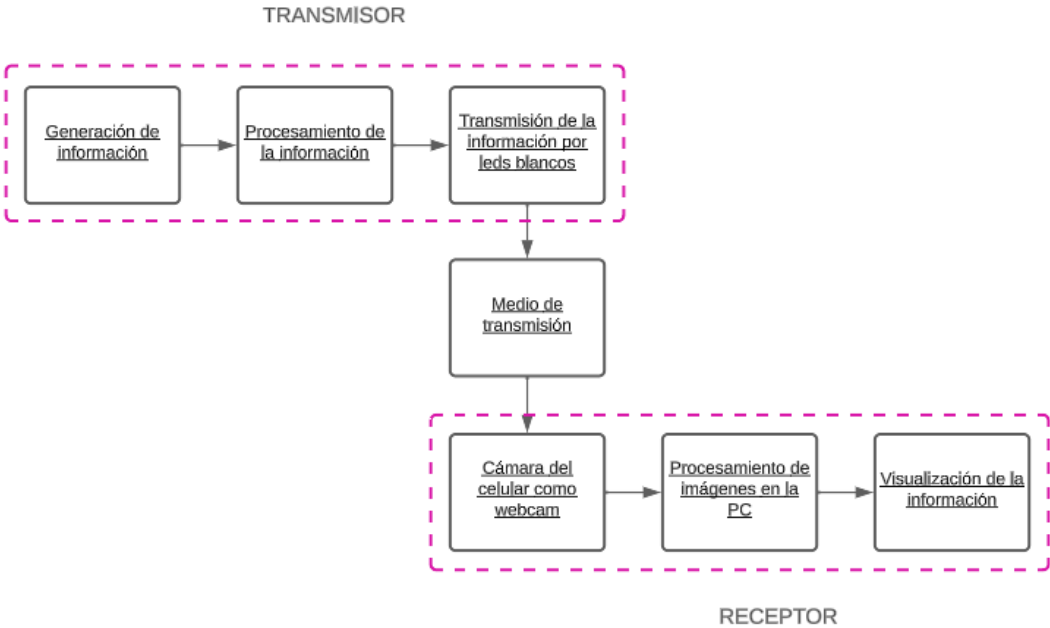
2.2 Escenarios propuestos

En esta sección se describen en detalle 2 posibles soluciones para la implementación del sistema OCC, representando situaciones hipotéticas y la metodología desarrollada. Cada una contendrá una descripción detallada de los elementos involucrados

2.2.1 Solución 1: Transmisión con modulación OOK, codificación Manchester y LEDs blancos.

En el siguiente bloque, se describe el sistema de transmisión que utiliza la modulación OOK, la codificación Manchester y los LEDs blancos. En la Figura 2, se muestra de forma visual la representación de estos bloques en el sistema, esta configuración permite la transmisión eficiente de datos binarios, donde la modulación OOK se encarga de variar la presencia y ausencia de una señal portadora para representar los bits, mientras que la codificación Manchester evita problemas de sincronización.

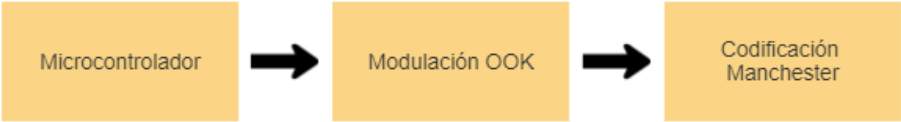
Figura 2
Sistema OCC con transmisión por LEDs blancos



Nota. En el diagrama se muestra el proceso de transmisión y recepción de un sistema OCC con LEDs blancos.

El bloque de *Generación de la información* representa los datos que se van a transmitir, en este caso serían los caracteres del 0 al 9. Por otro lado, se tiene el bloque de procesamiento de la información, el cual internamente se ve representado como se muestra en la Figura 3.

Figura 3
Procesamiento de la información



Nota. En la figura se muestra el diagrama interno del bloque de *Procesamiento de la información*.

Mediante un microcontrolador se procesó la información a transmitir, para ello se utilizó la modulación OOK, la cual se asemeja a un interruptor ya que sus valores lógicos oscilan entre 0 y 1, también se utilizó la codificación Manchester ya que es una manera factible de

codificar secuencias de bits, sobre todo cuando tienen un amplio periodo de duración. Para finalizar el sistema transmisor se tiene el bloque de *Transmisión de información por LEDs blancos*, en el cual, una vez que la información fue modulada, codificada y representada en encendidos y apagados del LED, se empieza el envío de datos por medio del aire. Para recibir dicha información se utilizó una cámara de celular como fotorreceptor, estas señales captadas fueron demoduladas y decodificadas con la ayuda de Python, el cual permite mostrar en pantalla los datos que se han transmitido luego de procesarlos e interpretarlos.

2.2.2 Solución 2: Transmisión con modulación PWM y LEDs RGB.

En la Figura 4 se muestra el proceso necesario para el tratamiento de la información proporcionada, en este caso por medio de un microcontrolador y con la ayuda de la modulación PWM, la cual permite modificar los ciclos de trabajo para una mejor representación de los datos en los LEDs RGB. En el desarrollo e implementación del código en el microcontrolador, se estableció con un rango de frecuencias de transmisión mayor que 100 Hz, superando el umbral perceptible por el ojo humano en lo que respecta al fenómeno del parpadeo visual. Dicha frecuencia se seleccionó con el objetivo de garantizar una percepción óptima, continua y sin interrupciones del mensaje generado por el LED RGB. Además, se ha diseñado un esquema de transmisión, el cual se apoya en la utilización de dos canales independientes: uno de ellos destinado a la transmisión de la luz roja y el otro orientado hacia la transmisión de la luz azul.

En lo referente a la recepción y decodificación de los datos, se implementó un algoritmo desarrollado en Python, representada en la Figura 5. El código hace uso de la biblioteca OpenCV, se encuentra apta con la capacidad y herramientas necesarias para llevar a cabo la demodulación precisa y efectiva de la señal transmitida por el LED, capturando y procesando el espectro de luz correspondiente. Las capturas de imágenes se realizarán

mediante la cámara del teléfono celular para luego ser procesadas en la PC con la ayuda del algoritmo en Python con previo desarrollo.

Figura 4

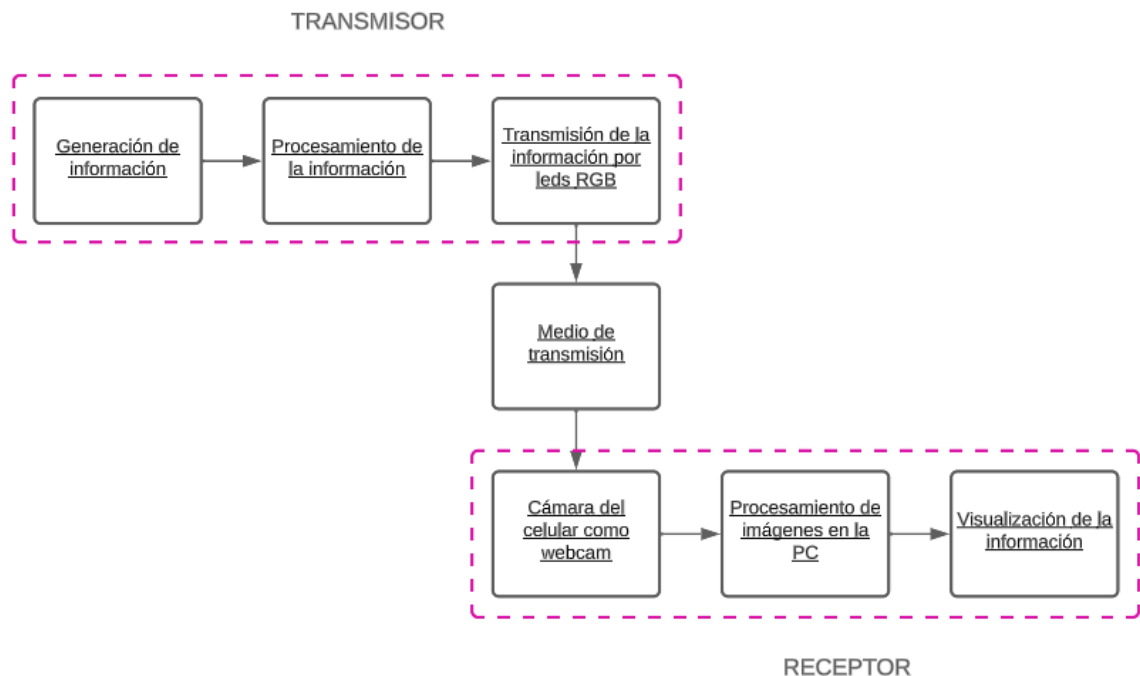
Procesamiento de la información



Nota. En la figura se muestra el diagrama interno del bloque de *Procesamiento de la información*.

Figura 5

Sistema OCC con transmisión por LEDs RGB



Nota. En el diagrama se muestra el proceso de transmisión y recepción de un sistema OCC con LEDs RGB.

Luego de haber presentado al cliente las dos soluciones planteadas anteriormente, se toma la decisión de implementar la solución 2.

2.3 Diagrama de flujo de la solución propuesta.

Luego de un minucioso análisis de las dos soluciones propuestas para el proyecto, se ha tomado la decisión de implementar la modulación PWM con luces RGB en lugar de la transmisión mediante modulación OOK, codificación Manchester y LEDs blancos. Esta elección se basa en diversas ventajas que ofrecen en términos de manipulación de los canales de comunicación para la transmisión de información. La modulación PWM con luces RGB proporciona un mayor control sobre la transmisión de datos al permitir variar el ciclo de trabajo de los colores rojo y azul. Esta flexibilidad es importante para adaptarse a diferentes necesidades de comunicación y optimizar la eficiencia de la transmisión. Cabe destacar que la distancia de transmisión requerida no supera los 20 cm. Esta corta distancia minimiza la posibilidad de pérdida de datos y garantiza una comunicación efectiva. Las Figuras 6 y 7 muestran los diagramas de flujo de la propuesta de solución seleccionada, al poder visualizarla de manera secuencial, se facilita el entendimiento e identificación de posibles mejoras al sistema OCC.

Figura 6

Diagrama de flujo del transmisor



Nota. En el diagrama se muestra el proceso del transmisor.

En la Figura 6, se muestra el proceso que se lleva a cabo para el funcionamiento del transmisor, en este caso se empezó definiendo las variables, una vez culminado el proceso anterior se configura la señal y el canal para realizar el envío de la información, la cual se analiza y se entrega el mismo dato definido como *char*. El carácter entregado es comparado desde la posición 3 hasta la posición 0, verificando que el número enviado coincida en formato binario, finalmente, dependiendo de cada carácter se asigna un ciclo de trabajo por cada canal.

En el diagrama de flujo presentado en la Figura 7, se muestra el proceso al que se somete el receptor, el cual se encarga de utilizar la cámara del teléfono como

webcam para percibir las señales de luz e interpretarlas de acuerdo con lo definido en el transmisor, finalmente se muestra en pantalla dichos datos.

Figura 7

Diagrama de flujo del receptor



Nota. En el diagrama se muestra el proceso del receptor

CAPÍTULO 3

3 RESULTADOS Y ANÁLISIS

En relación con lo analizado en el Capítulo 2, se llevaron a cabo las pruebas de funcionamiento correspondientes a nuestro sistema de comunicación. Estas pruebas se presentarán detalladamente en el presente capítulo, acompañadas de un análisis exhaustivo de los parámetros inherentes a nuestra comunicación y su grado de eficacia. El propósito es respaldar empíricamente el correcto desempeño de nuestro sistema, a través de resultados concretos.

3.1 Implementación de la solución propuesta

3.1.1 Transmisor

El código del transmisor implementa un sistema de control de LEDs mediante el uso de modulación PWM y comunicación serial. En esencia, permite ajustar el brillo de dos LEDs (rojo y azul) en función de los números recibidos a través del puerto serial.

Cuando se ejecuta el programa, se inicia la comunicación serial a una velocidad de 9600 bps, lo que permite la interacción con el usuario a través del monitor serial. Luego, se configuran los parámetros de PWM para los LEDs utilizando la función **ledcSetup**, y se asocian los pines físicos a los canales PWM mediante **ledcAttachPin**.

En el bucle principal (**loop**), el programa espera a que lleguen datos por el puerto serial.

Cuando se recibe un número dentro del rango válido (0-15), se utiliza una función de codificación para asignar valores de ciclo de trabajo específicos a los LEDs rojo y azul. Estos valores determinan la intensidad luminosa de los LEDs, ajustando su brillo.

Una vez calculados los valores de ciclo de trabajo, se utilizan las funciones **ledcWrite** para actualizar el brillo de los LEDs en función de los valores calculados anteriormente. Esto permite que los LEDs cambien su intensidad de luminosidad en tiempo real según el número recibido.

Si el número recibido no está dentro del rango válido, se muestra un mensaje de error en el monitor serial indicando que el número está fuera del rango permitido.

Con respecto al análisis anterior, se decidió hacer uso único de los colores rojo y azul como canales de transmisión. Los colores rojo y azul son componentes independientes de la luz visible, lo que permite la creación de dos canales de transmisión que no interfieren entre sí. Esto mejora la capacidad de transmitir datos de manera simultánea y eficiente, además, estos colores se encuentran en partes del espectro luminoso que son menos propensas a la interferencia y atenuación en ciertos entornos, resultando en una mayor eficiencia de transmisión y menos pérdida de datos dado que al analizar los ciclos de trabajo de los colores rojo y azul, es posible lograr una interpretación más precisa de la información transmitida y su combinación proporciona redundancia y ayuda a mitigar posibles errores de transmisión.

3.1.2 Receptor

Si bien el transmisor no varía, el receptor pasó por varios cambios hasta encontrar una solución definitiva. A continuación, se presenta el proceso llevado a cabo.

Pruebas con cámara web.

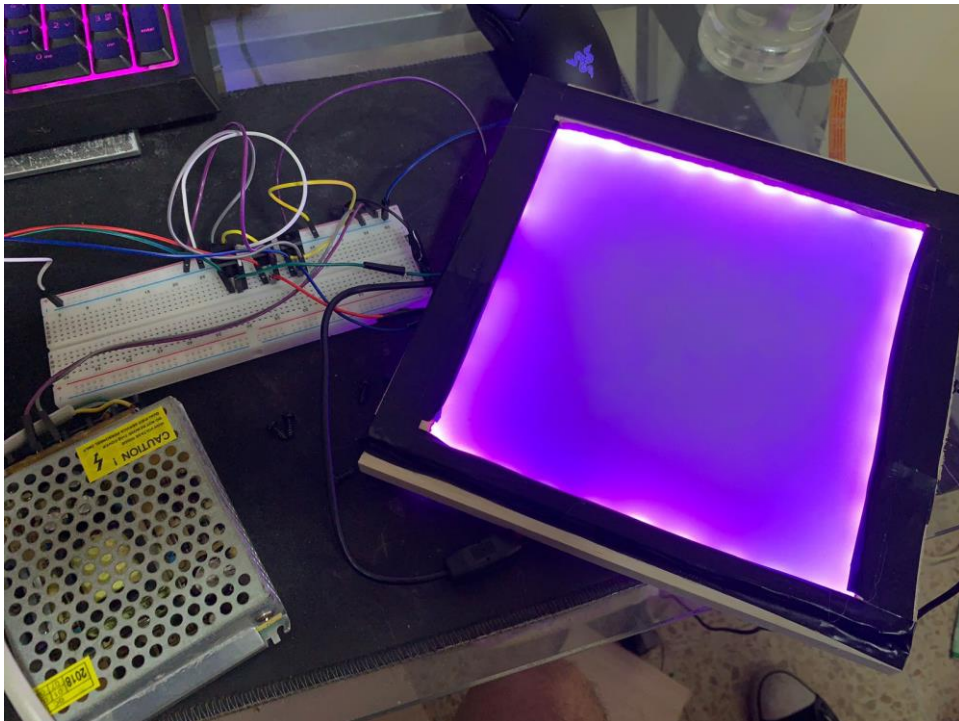
Tal como se muestra en la Figura 8, el prototipo diseñado consistió en la utilización de un microcontrolador Arduino Mega como transmisor, configurado para generar señales de modulación PWM con diferentes ciclos de trabajo, los cuales representaban valores que variaban desde 10 hasta 255. Esta variación en los ciclos de trabajo permitía lograr una modulación de la intensidad de las luces en la tira LED RGB. Para asegurar una adecuada modulación, se estableció una frecuencia de señal de 240 Hz mediante un delay de 4000 us en la configuración del Arduino. Esto se debe a que el espectro de luz visible muestra que el color verde se superpone considerablemente con los otros dos colores, lo que podría potencialmente causar interferencias en nuestra transmisión. Al optar por utilizar solo los colores rojo y azul, estamos optimizando la eficacia de la comunicación y minimizando las posibles complicaciones derivadas de las superposiciones en el espectro de luz visible. Se emplearon tres transistores mosfet de tipo canal N como interruptores. Estos dispositivos actuaron como una especie de llave electrónica, permitiendo encender y apagar los LEDs de manera rápida y precisa. Al aplicar la señal PWM generada por el Arduino a los mosfets, se logró controlar el flujo de

corriente a través de los LEDs y, en consecuencia, variar la intensidad lumínica emitida por la tira.

Cabe destacar que la fuente de alimentación utilizada para el sistema fue de 12 V, la cual proporcionó la energía necesaria tanto para la tira LED como para los componentes electrónicos involucrados. Por otro lado, en el receptor se implementó un programa en Python que se encargaba de interpretar la señal transmitida por el Arduino. Para lograr esto, se hizo uso de la cámara web del sistema, la cual capturaba los ciclos de trabajo generados por el Arduino. Mediante un algoritmo de procesamiento de imágenes, el programa en Python fue capaz de determinar los números correspondientes a los ciclos de trabajo y, en consecuencia, ajustar la intensidad de la tira LED de manera precisa y en tiempo real.

Figura 8

Prototipo inicial del sistema OCC.



Nota. En la imagen se muestra el prototipo de prueba.

En las Figuras 9 y 10, se pueden apreciar las señales moduladas correspondientes a los canales rojo y azul, respectivamente, que fueron medidas a través de un osciloscopio. En estas mediciones, se observa la variación del ciclo de trabajo de ambos canales, con un 47%

para el canal rojo y un 53% para el canal azul. Estos valores fueron configurados en el software de Arduino para enviar datos específicos, sin embargo, durante el proceso de transmisión de las señales, se detectó un problema crítico: las señales se invertían, lo que resultaba en una incorrecta transmisión de los datos. Después de una detallada revisión y análisis, se identificó que los errores residían en la programación del Arduino, las cuales fueron corregidas en la implementación final.

Figura 9

Señal transmitida en el canal rojo.



Nota. En la imagen se puede visualizar la señal PWM transmitida en el canal rojo.

Figura 10

Señal transmitida en el canal azul.



Nota. En la imagen se puede visualizar la señal transmitida por el canal azul

Al examinar detenidamente la Figura 11, se puede apreciar claramente las líneas que representan los ciclos de trabajo de nuestras señales PWM. El grosor de estas líneas se

correlaciona directamente con el porcentaje de trabajo de cada una de las señales. Sin embargo, como se señaló en la sección anterior, lamentablemente se encuentra con un inconveniente en la transmisión de datos que afectó negativamente la correcta definición de estas líneas.

Este desafortunado problema de transmisión de datos tuvo un impacto significativo en la recepción del dato leído por la cámara. Como resultado, la precisión de la información transmitida a través de las señales PWM se vio comprometida, lo que condujo a una incorrecta interpretación de los ciclos de trabajo y, en última instancia, a una mala lectura por parte de la cámara.

Figura 11

Lámpara vista desde la cámara web.



Nota. En la imagen se puede visualizar la lámpara vista desde un cámara web.

Pruebas con aplicación móvil ImagePro

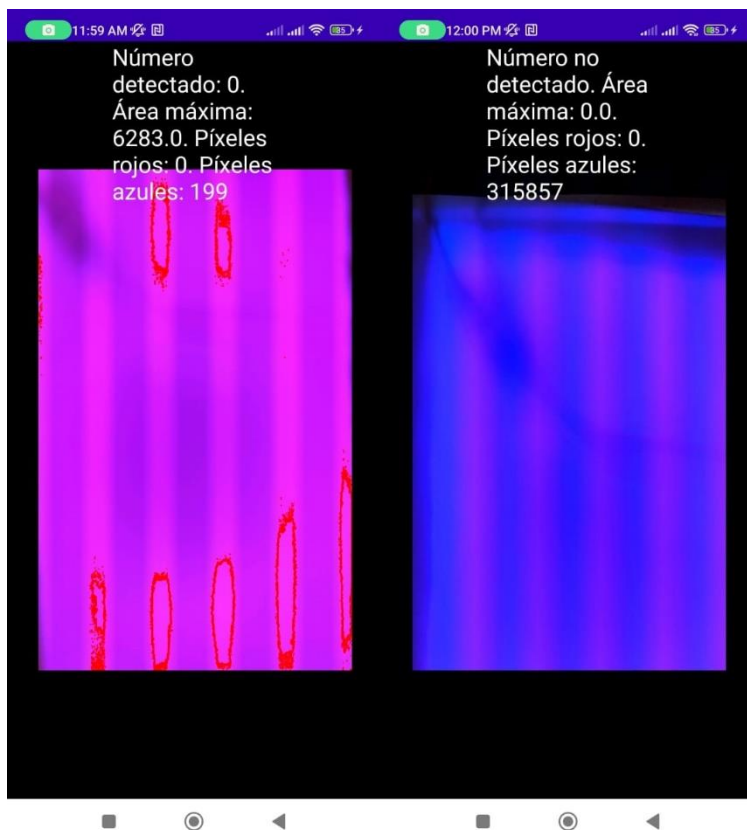
Las pruebas realizadas de manera directa con el teléfono celular se hicieron con una aplicación móvil base, en este caso se utilizó ImagePro, la cual tiene integrada la librería OpenCV, sobre esta aplicación se introdujo código que permitiera llevar a cabo el procesamiento de imágenes en tiempo real. La imagen capturada por la cámara es llevada a

escala de grises para luego separar los pixeles iluminados de los que no lo están, se intenta decodificar los números de acuerdo con la región de pixeles iluminados, para ello se dibuja el contorno más grande encontrado y se compara con el área máxima agregada a cada número.

Una vez ejecutada la aplicación en el celular se puede observar en la Figura 12 que decodifica únicamente el número 0, ya que como se explicó anteriormente, el ancho de las líneas depende del ciclo de trabajo asignado a cada carácter, por lo cual, mientras más tenues sean estas líneas, la aplicación no podrá bordearlas y por ende no detectar los números que se están transmitiendo.

Figura 12

Aplicación móvil desarrollada en base a ImagePro



Nota. En la imagen se muestra la decodificación del cero y el fallo en los demás números.

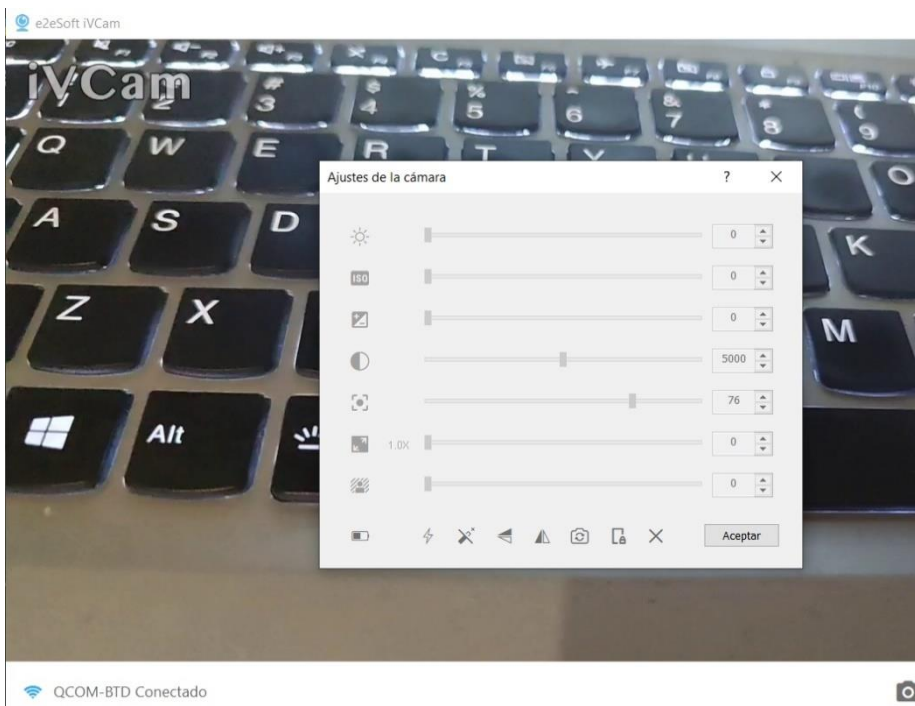
Pruebas con celular como cámara web

Para mejorar el funcionamiento del sistema OCC, se utilizó la cámara del teléfono como cámara web con la ayuda de la aplicación iVCam y las imágenes capturadas se procesaron en una PC en el software Visual Studio Code y Python como lenguaje de programación. Como se

muestra en la Figura 13, para poder visualizar las franjas de colores es necesario modificar ciertos parámetros de la cámara, para ello es necesario acceder al panel de configuraciones de la aplicación en la Pc y modificarlos, en este caso la exposición se configura a 0 y el parámetro ISO varía dependiendo de la iluminación a la que se está trabajando, para las pruebas de lo configuró en 500. En la Figura 14 se muestra como la transmisión del número 0, los demás números se pueden observar en el anexo A.

Figura 13

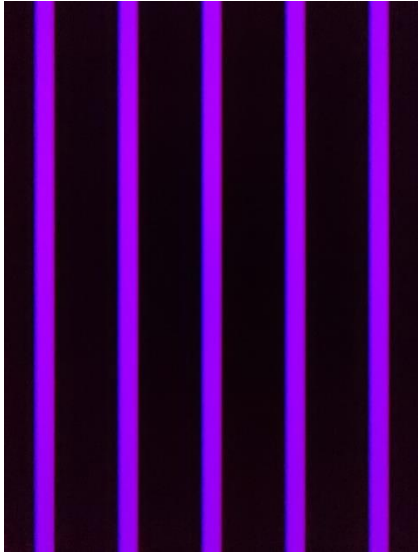
Ajustes de cámara.



Nota. En la imagen se muestra las opciones de ajuste de cámara en iVCam.

Figura 14

Transmisión del "0" visto desde la cámara del teléfono.



Nota. En la imagen se muestra la transmisión del 0.

Una vez configurada la cámara, se procede al análisis y procesamiento de las imágenes capturadas en tiempo real, para ello se empieza separando los canales rojo y azul con la ayuda de la función `cv2.split`, los cuales entregan una imagen en escala de grises. Cada píxel recibe un valor de umbral, 180 para el azul y 60 para el rojo, estos valores pueden variar de acuerdo con la iluminación del lugar en que se realizan las pruebas, si el píxel tiene un valor inferior al umbral, se convierte en 0; caso contrario, se le da un valor máximo.

Posteriormente, se efectúa una cuantificación de color que reduce los colores presentes en una imagen. A veces, algunos dispositivos tienen limitaciones y solo pueden representar una cantidad específica de colores. Es crucial convertir la imagen en una matriz de tamaño $M \times 3$, siendo M el total de píxeles en la imagen. Una vez agrupados, otorgamos valores centrales a cada píxel, logrando que la imagen resultante tenga una cantidad definida de colores. Como ejemplo, si deseamos que contenga únicamente 2 colores, fijamos $K=2$. Posteriormente, se debe retornar la imagen a su estructura inicial, al final se podrá observar la imagen binaria tal como se muestra en la Figura 15.

Figura 15

Imagen binaria.

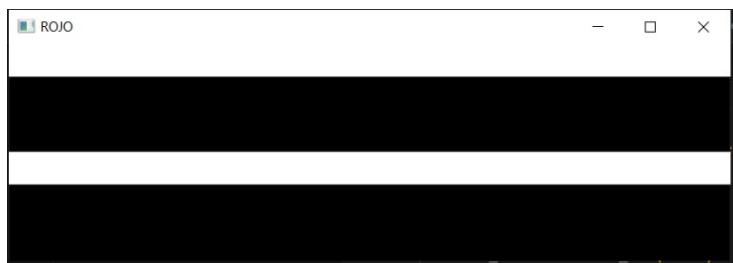


Nota. Se muestra la imagen reconstruida solo con 2 colores.

Los contornos son esenciales para analizar formas, detectar y reconocer objetos. En OpenCV, identificar contornos se asemeja a localizar un objeto blanco sobre un fondo oscuro. Por ello, se adaptó la imagen a solo dos colores, simplificando así la búsqueda de estos contornos, además, para recortar la imagen, así como se aprecia en la Figura 16, se toma en cuenta desde el inicio del segundo contorno hasta el inicio del último contorno, esto se realiza para tener una imagen completa del número transmitido. (*OpenCV: Contours : Getting started*, s. f.)

Figura 16

Imagen binaria recortada.



Nota. Se muestra la imagen binaria recortada.

La imagen binaria recortada se ingresa en la función `porcentaje_valores_pixeles`, la cual como indica su nombre, con la ayuda de la función `cv2.countNonZero` se calculan los porcentajes de píxeles de fondo y objeto de cada canal. Estos valores son utilizados para la decodificación del símbolo, función en la cual se indica que el valor de los ciclos de trabajo tiene un margen de $\pm 10\%$ de su valor original.

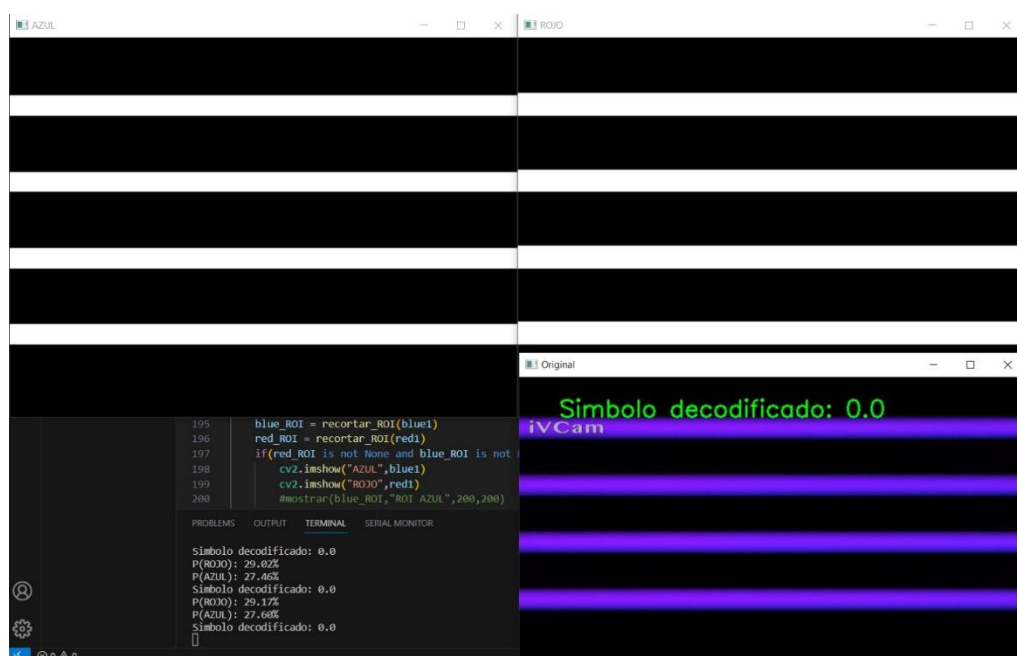
En el Anexo A se pueden apreciar las imágenes de los números 24 al 38 donde se encuentra la transmisión de datos referente a todos los caracteres que es posible enviar en nuestro sistema. Estos datos son representados mediante la variación del ciclo de trabajo de las luces RGB donde cada carácter representa un porcentaje diferente, en la pantalla se aprecia el cambio del ancho de las líneas dependiendo del carácter enviado.

El porcentaje del ciclo de trabajo de los caracteres decodificados y su correcta recepción se muestra en la terminal del programa Visual Studio Code tal como se observa en la Figura 17 y en el Anexo B con las imágenes de los números 39 al 53.

Finalmente, tal como se muestra en la Figura 17, el número decodificado es visualizado tanto en la pantalla de la PC como en la pantalla del teléfono, todas las pruebas con los demás números se pueden apreciar en el anexo B. En la Figura 18 se muestra la manera en la que el número decodificado se recibe en el teléfono celular, esto se debe a que se utilizó la aplicación móvil Pushbullet, la cual genera un token que permite enviar cualquier información a manera de notificaciones.

Figura 17

Decodificación del "0"



Nota. En la imagen se muestra el número 0 procesado y decodificado.

Figura 18

Visualización del número decodificado en el teléfono celular.

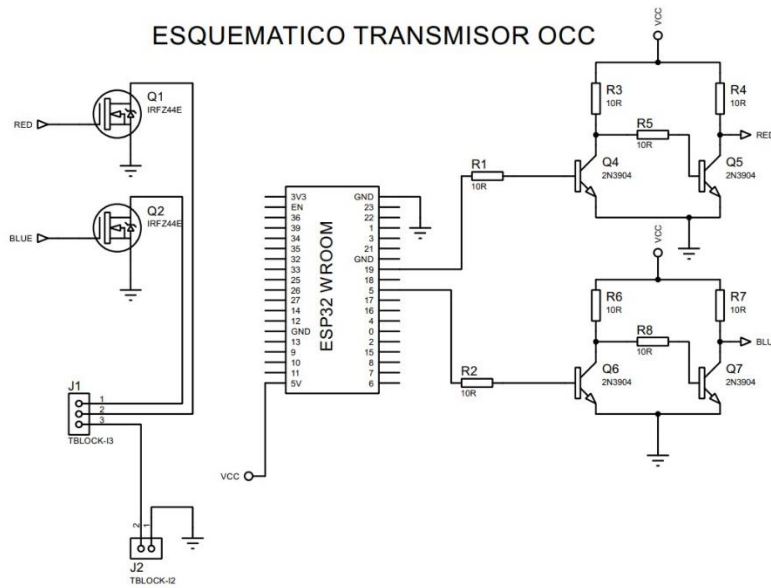


Nota. En la figura se muestra una notificación del número transmitido.

En la Figura 19 y la Figura 20, se presenta el diseño definitivo del esquemático correspondiente al transmisor. En esta configuración, se ha incorporado una placa programable ESP32, que, en comparación con el Arduino Mega, ostenta la capacidad de procesar información a una velocidad superior. Un aspecto para mencionar es la inclusión de un conjunto de transistores en cada canal, la finalidad de esta adición radica en la elevación del voltaje en la entrada de los MOSFET, lo que a su vez facilita la inducción de la saturación de estos. Como resultado de esta operación, se logra su conversión en interruptores eficientes, permitiendo el desempeño óptimo de su papel. Es relevante destacar que este arreglo circuital ha sido implementado físicamente en una placa PCB, valiéndose para ello del entorno de desarrollo proporcionado por el software Proteus.

Figura 19

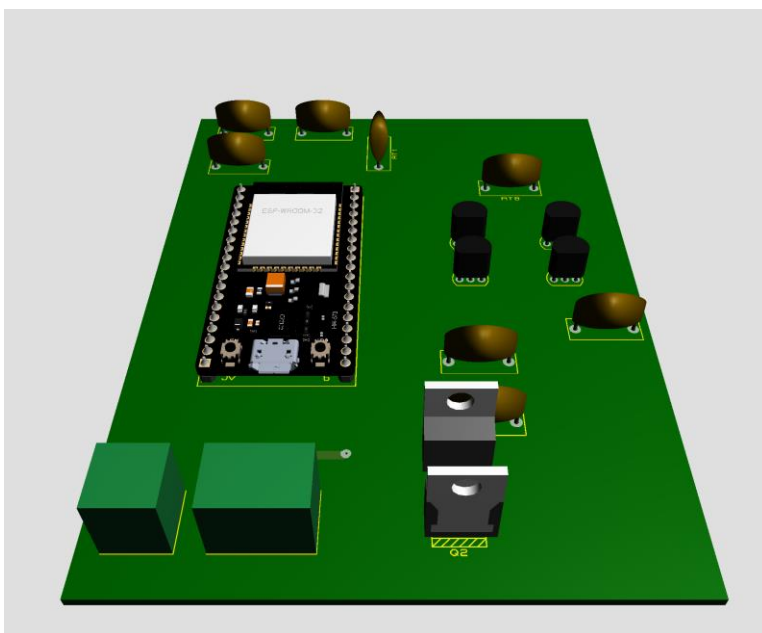
Esquemático modificado



Nota. En la imagen se muestra el esquemático modificado.

Figura 20

Diagrama esquemático final



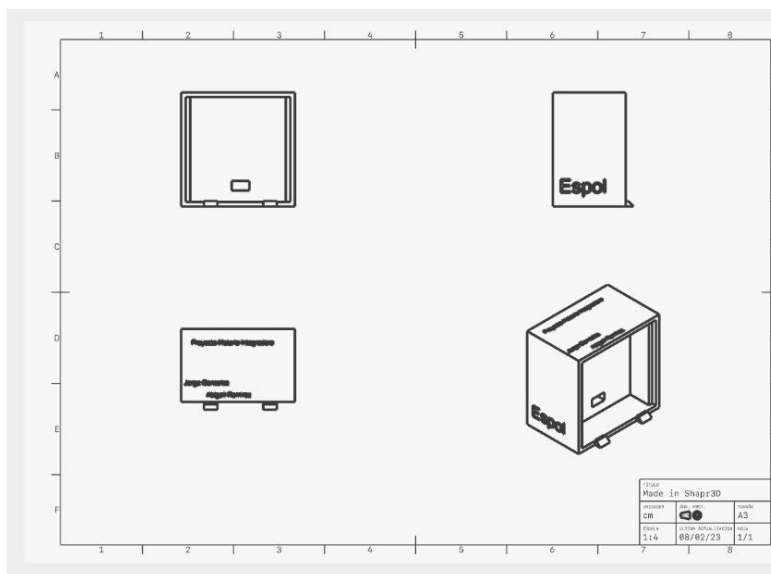
Nota. En la imagen se muestra el diagrama esquemático del prototipo modificado.

La Figura 21 muestra el diseño definitivo del transmisor, que ha sido representado en un plano detallado. En este esquema, se destaca la disposición de los componentes clave, como el microcontrolador y la tira LED que vamos a utilizar. Además, hemos incorporado un difusor en esta lámpara, diseñado utilizando el programa Sharp 3D, para mejorar la apreciación de los colores emitidos.

La caja que alberga estos elementos posee dimensiones precisas de 25x25x10 cm, lo que garantiza un espacio adecuado para acomodar tanto los componentes internos como el espacio necesario para la dispersión uniforme de la luz.

Figura 21

Diseño de caja



Nota. En la imagen se muestra el diseño de la caja del prototipo.

Prueba final con el sistema OCC implementado.

El sistema OCC final consta de la carcasa diseñada, un teléfono celular y una PC, la cual es utilizada para el procesamiento de imágenes. El celular es colocado en la base a una distancia de 20 cm para recibir cada carácter que se envíe a través de la lámpara integrada en la carcasa.

Para empezar a detectar la información transmitida es necesario abrir la aplicación móvil IVCam, la cual permitirá que la PC acceda de manera remota a la cámara del teléfono y así

ajustar los parámetros necesarios para la visualización de las líneas que se muestran en la Figura 22.

Figura 22

Sistema OCC final



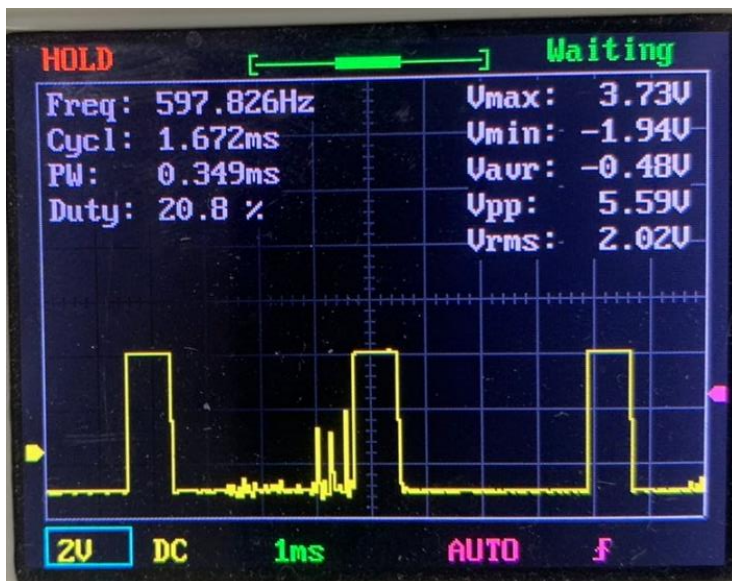
Nota. En la imagen se muestra la implementación final del sistema OCC.

La visualización de la señal PWM en la Figura 23 proporciona una mejor comprensión de la técnica de modulación utilizada para transmitir información, vista desde un osciloscopio. La elección de un ciclo de trabajo del 20% para representar el dígito "0" no solo demuestra la eficiencia en la asignación de caracteres, sino que también sugiere una cuidadosa optimización del sistema. Este ciclo de trabajo específico, que establece la relación entre la duración de los pulsos y los espacios en la señal, presenta una serie de implicaciones. Por un lado, denota la

capacidad del sistema para transmitir otros caracteres a través de la variación del ciclo de trabajo. Además, esta selección indica una consideración de la resistencia al ruido, ya que el ciclo elegido podría conferir una mayor inmunidad a posibles interferencias. Esta elección también destaca la necesidad de una rigurosa sincronización entre el emisor y el receptor, asegurando una interpretación precisa de los datos. En última instancia, la observación de este parámetro no solo confirma la efectividad de la técnica de modulación, sino que también revela un diseño sistemático y una planificación cuidadosa detrás del proceso de transmisión de datos

Figura 23

Señal PWM canal rojo.



Nota. En la imagen se puede visualizar la señal PWM transmitida en el canal rojo.

Para las pruebas se enviaron 5 grupos de 10 datos, tal como se muestra en la tabla 2 del anexo C, teniendo en cuenta que los caracteres solicitados por el cliente son del 0 al 9; se pudo observar que existe un error de 4% ya que, en un total de 50 caracteres enviados, 2 de ellos se recibieron de manera incorrecta, como es el caso del número 0, el cual en dos ocasiones se recibió el número 1. Este tipo de inconvenientes puede presentarse debido a la iluminación y a los parámetros de la cámara que hay que modificar como la exposición e ISO.

A continuación, se detalla el costo total de la implementación del sistema OCC en la tabla 1, en la cual se puede apreciar es de \$67.30.

Tabla 1

Tabla de costos de implementación del sistema OCC

Componentes	Unidades	Precio
2N3904	4	\$0.40
Resistor 1kOhm	4	\$0.40
Resistor 10kOhm	4	\$0.40
IRF360	2	\$3.60
ESP32 nodemcu	1	\$12
Baquelita	1	\$1.50
cloruro férrico anhidrido	3	\$6
Carcasa	1	\$28
Tira LED RGB	1	\$15
TOTAL		\$67.30

Nota. Detalles de precios de los componentes del sistema OCC.

CAPÍTULO 4

4 CONCLUSIONES Y RECOMENDACIONES

En esta sección se resaltan los resultados obtenidos mediante las pruebas realizadas y detalladas en el capítulo anterior, dando énfasis en varios puntos específicos

4.1 Conclusiones

- Los sistemas OCC pueden ser implementados con diferentes técnicas de modulación, se analizaron 2 de ellas como propuesta de solución, ofreciendo al cliente incorporar este sistema con la ayuda de la modulación OOK o PWM, siendo esta última la escogida.
- En base a las pruebas realizadas, tal como se muestra en la sección 3.1, se diseñó un sistema OCC capaz de recibir datos a una distancia de 20 cm con un error de 10^{-1} , el cual es calculado mediante el envío de 5 conjuntos de 10 números entre el 0 al 9.
- Con la ayuda del aplicativo móvil Pushbullet se logra visualizar en la pantalla del teléfono celular los datos recibidos y decodificados tal como se muestra en la Figura 18, permitiendo que una interacción agradable entre el sistema y el usuario.
- Se llevó a cabo un análisis acerca del procedimiento y manipulación del sistema OCC, definiendo los parámetros necesarios que se deben modificar para tener una mayor eficiencia en la transmisión y recepción de datos.
- Conforme se detalla en la sección 3.2, se llevó a cabo un análisis de costos, revelando que el costo total de la implementación de este proyecto asciende a \$67.30 dólares.

4.2 Recomendaciones

- Es recomendable definir un teléfono celular destinado a las pruebas, ya que los parámetros varían dependiendo de varios factores, entre ellos está la iluminación y la cámara utilizada.
- Para obtener un mejor resultado en la decodificación se recomienda recordar la imagen, ya que es mejor tener contornos completos a la hora de filtrar las capturas obtenidas por la cámara.
- Se recomienda el uso de la librería OpenCv para facilitar el procesamiento de imágenes, ya que esta cuenta una amplia variedad de funciones entorno a lo solicitado para este proyecto.

4.3 Trabajos Futuros

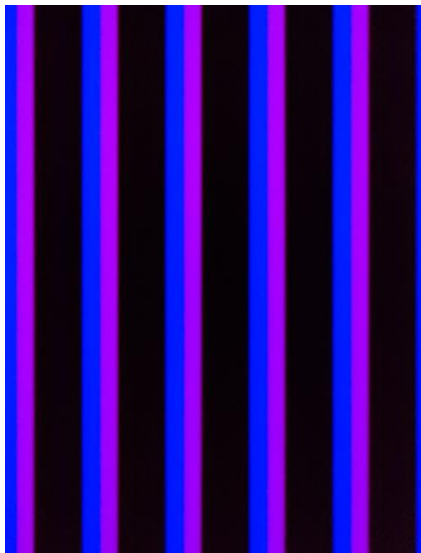
- Desarrollar una aplicación móvil que permita procesar las imágenes en tiempo real y se muestre en la pantalla del mismo los resultados obtenidos, con el fin mejorar la interacción con el sistema implementado.
- Cambiar el diseño de la carcasa con el fin de mejorar la lámpara para obtener un mejor alcance de transmisión y recepción de datos.

ANEXOS

Anexo A. Transmisión de datos. imágenes de los números 24 al 38

Figura 2424

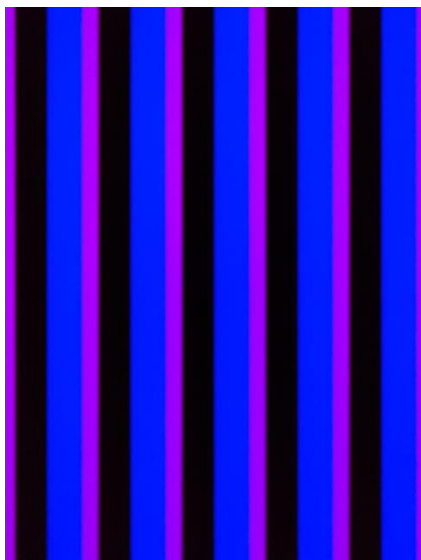
Transmisión del "1" visto desde la cámara del teléfono.



Nota. En la imagen se muestra la transmisión del 1.

Figura 2525

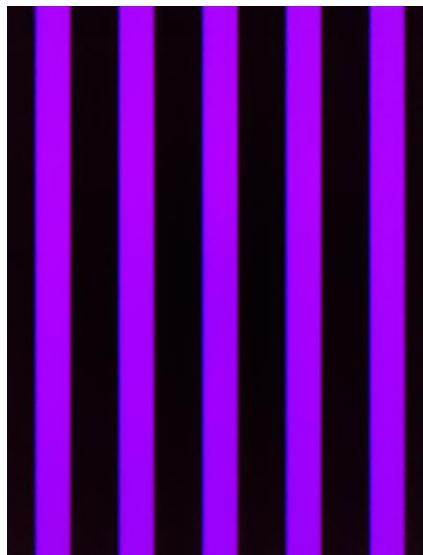
Transmisión del "2" visto desde la cámara del teléfono.



Nota. En la imagen se muestra la transmisión del 2.

Figura 2828

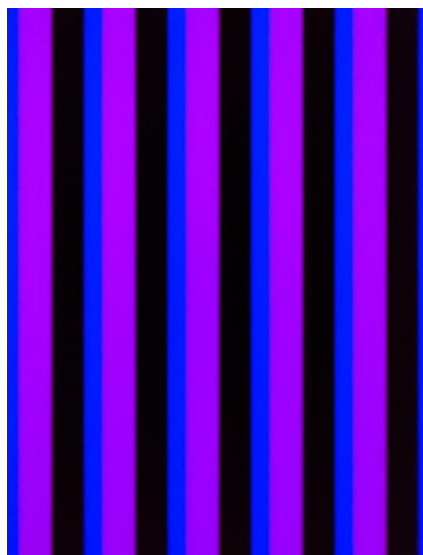
Transmisión del "5" visto desde la cámara del teléfono.



Nota. En la imagen se muestra la transmisión del 5.

Figura 2929

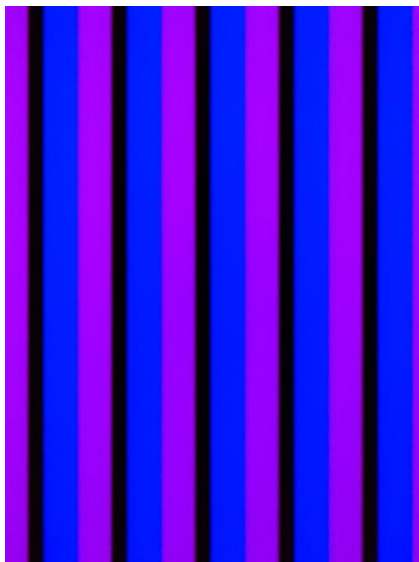
Transmisión del "6" visto desde la cámara del teléfono.



Nota. En la imagen se muestra la transmisión del 6.

Figura 3030

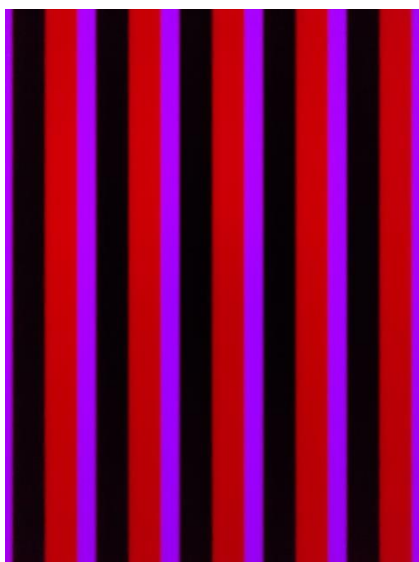
Transmisión del "7" visto desde la cámara del teléfono.



Nota. En la imagen se muestra la transmisión del 7.

Figura 3131

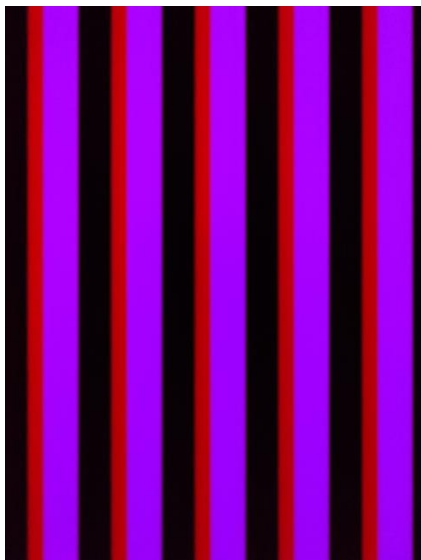
Transmisión del "8" visto desde la cámara del teléfono.



Nota. En la imagen se muestra la transmisión del 8.

Figura 3232

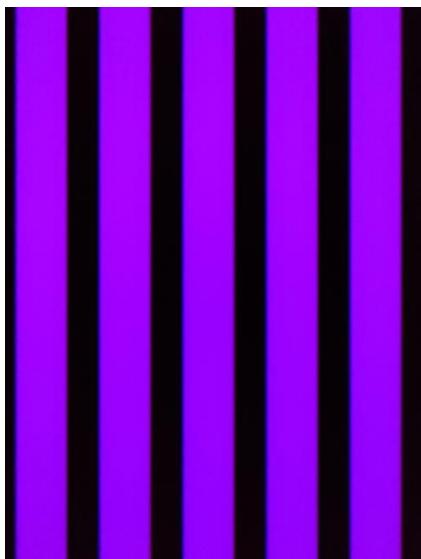
Transmisión del "9" visto desde la cámara del teléfono.



Nota. En la imagen se muestra la transmisión del 9.

Figura 3333

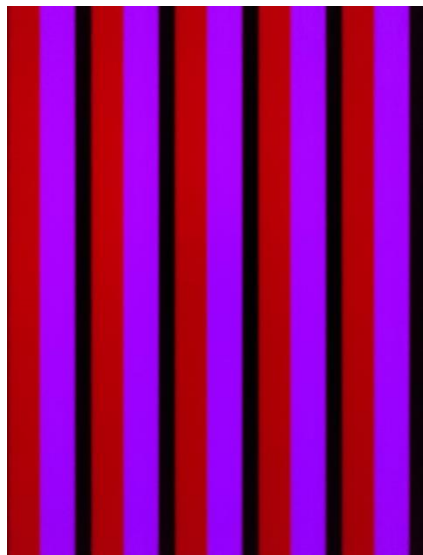
Transmisión del "10" visto desde la cámara del teléfono.



Nota. En la imagen se muestra la transmisión del 10.

Figura 3636

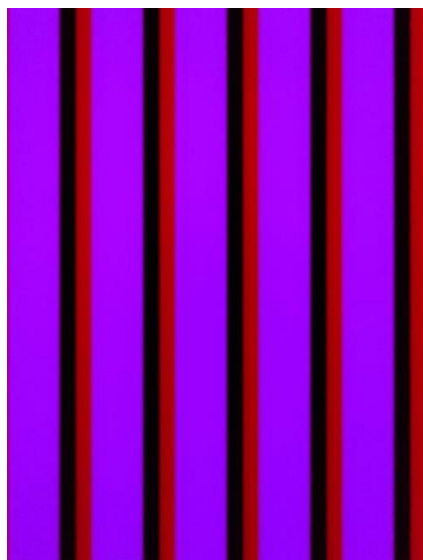
Transmisión del "13" visto desde la cámara del teléfono.



Nota. En la imagen se muestra la transmisión del 13.

Figura 3737

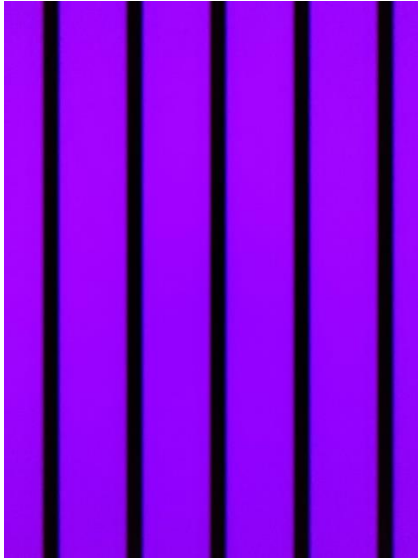
Transmisión del "14" visto desde la cámara del teléfono.



Nota. En la imagen se muestra la transmisión del 14.

Figura 3838

Transmisión del "15" visto desde la cámara del teléfono.

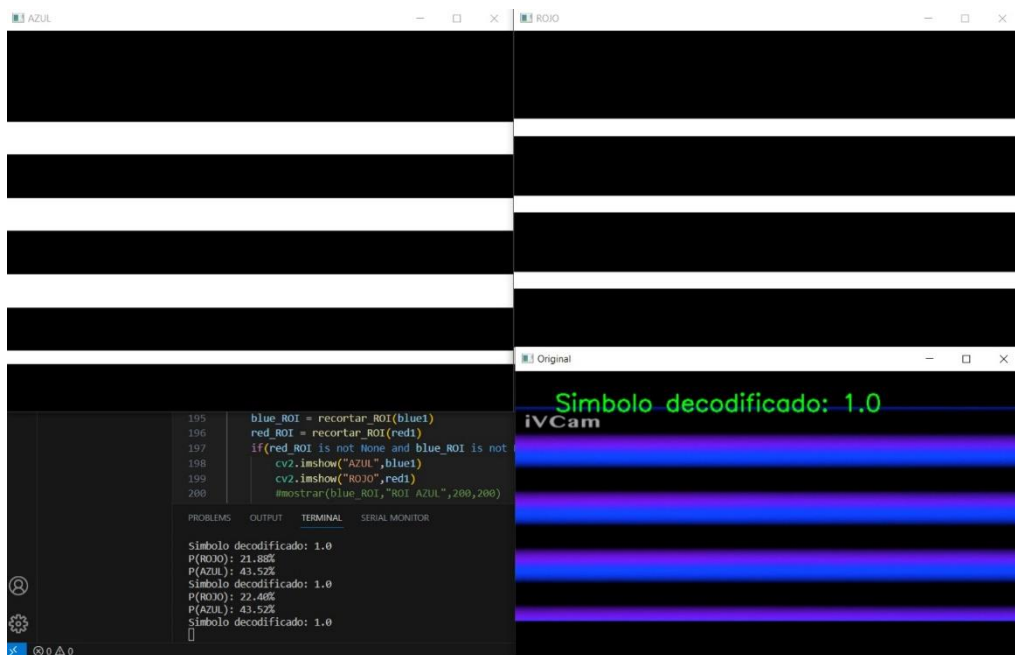


Nota. En la imagen se muestra la transmisión del 15.

Anexo B. Decodificación de datos. Imágenes de los números 39 al 53

Figura 3939

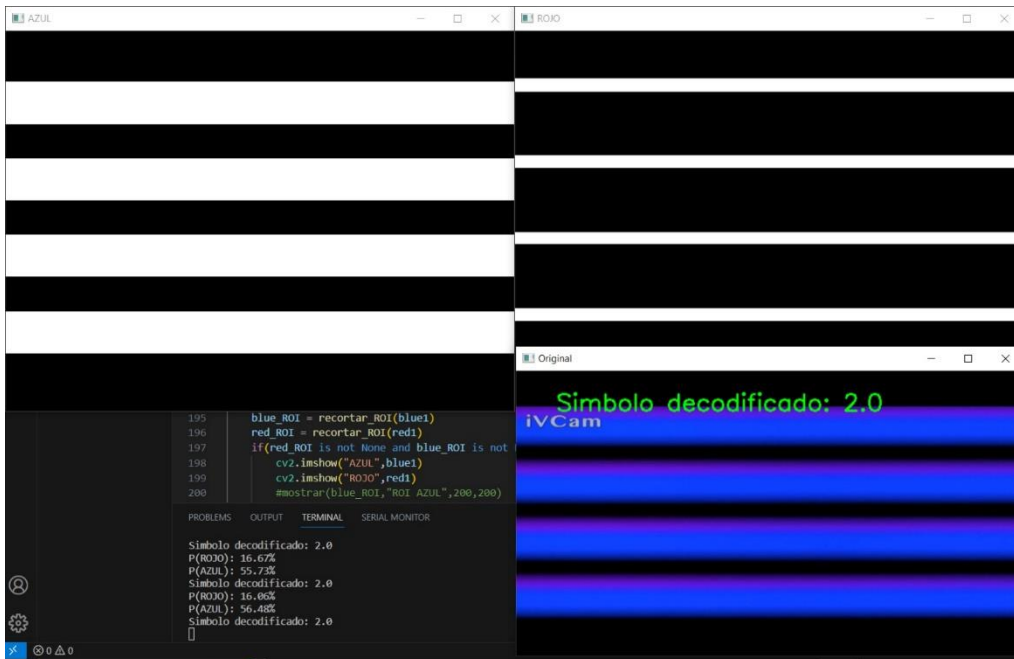
Decodificación del "1"



Nota. En la imagen se muestra el número 1 procesado y decodificado.

Figura 4040

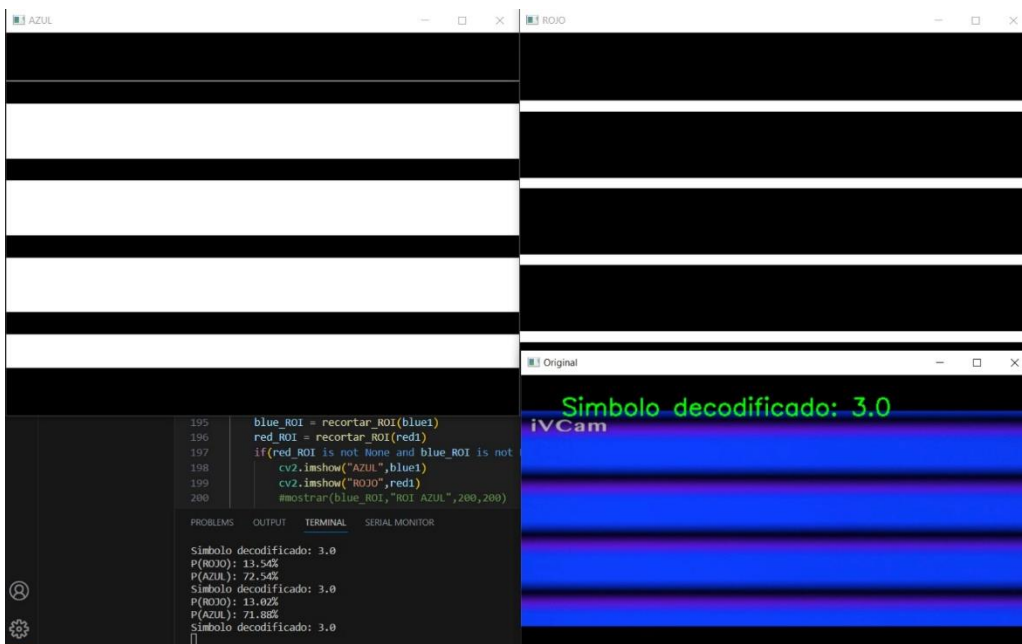
Decodificación del "2"



Nota. En la imagen se muestra el número 2 procesado y decodificado.

Figura 4141

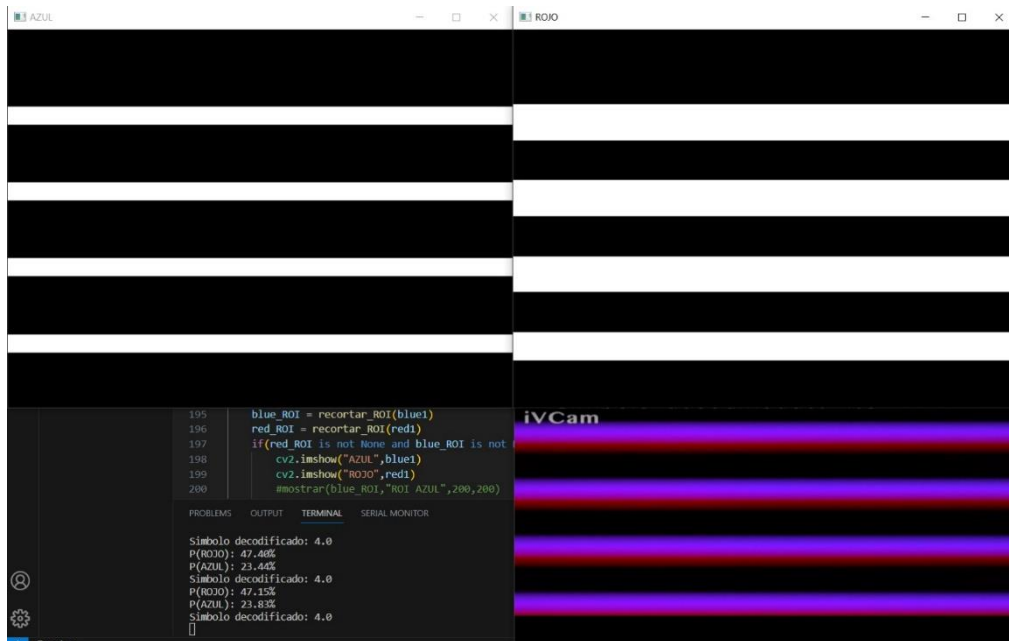
Decodificación del "3"



Nota. En la imagen se muestra el número 3 procesado y decodificado.

Figura 4242

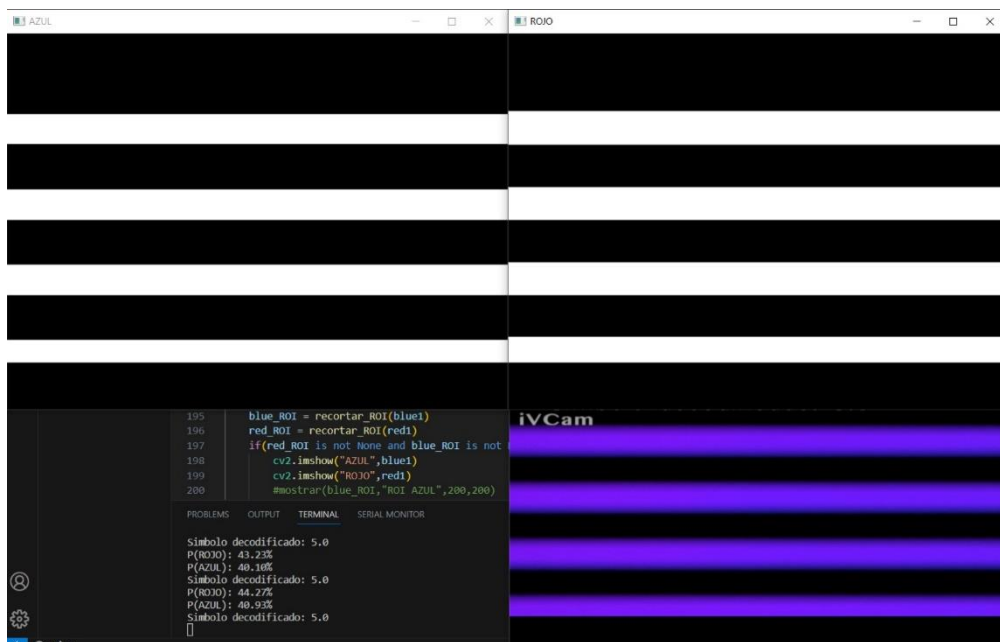
Decodificación del "4"



Nota. En la imagen se muestra el número 4 procesado y decodificado.

Figura 4343

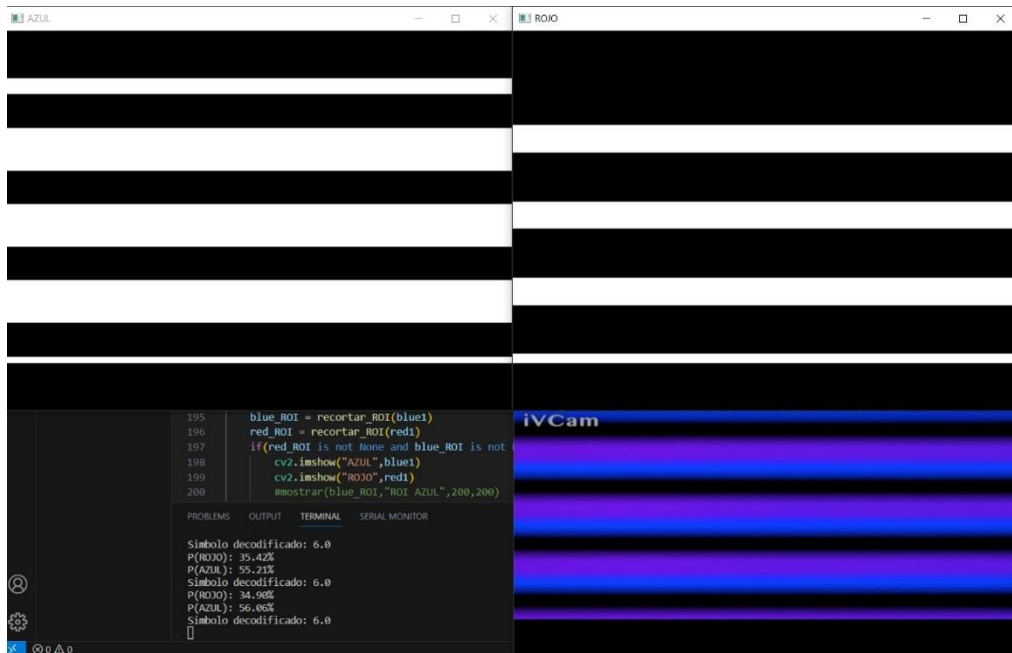
Decodificación del "5"



Nota. En la imagen se muestra el número 5 procesado y decodificado.

Figura 4444

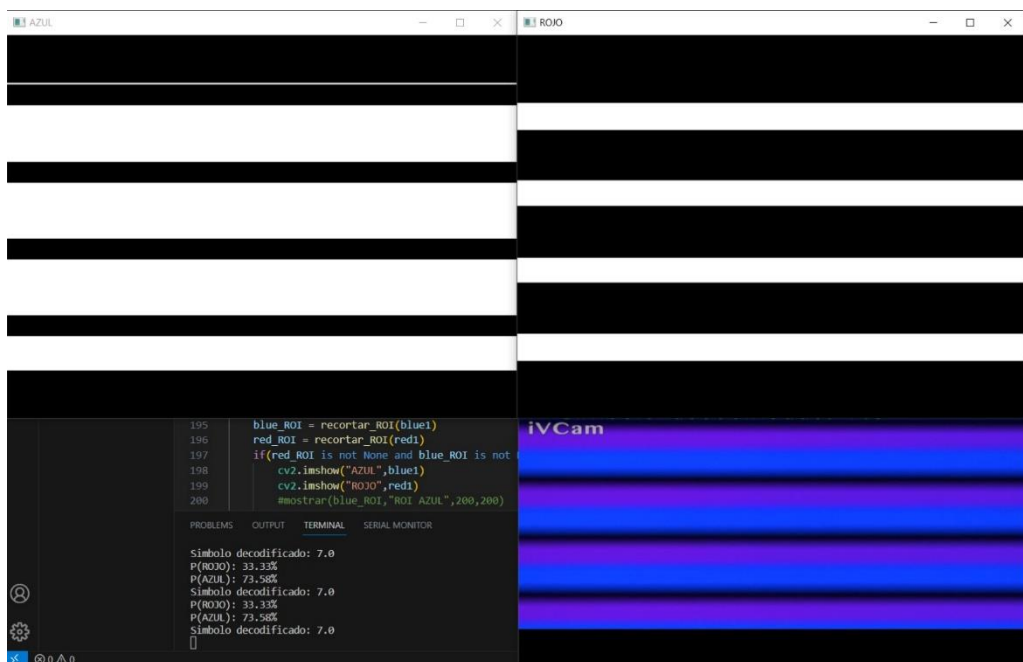
Decodificación del "6"



Nota. En la imagen se muestra el número 6 procesado y decodificado.

Figura 4545

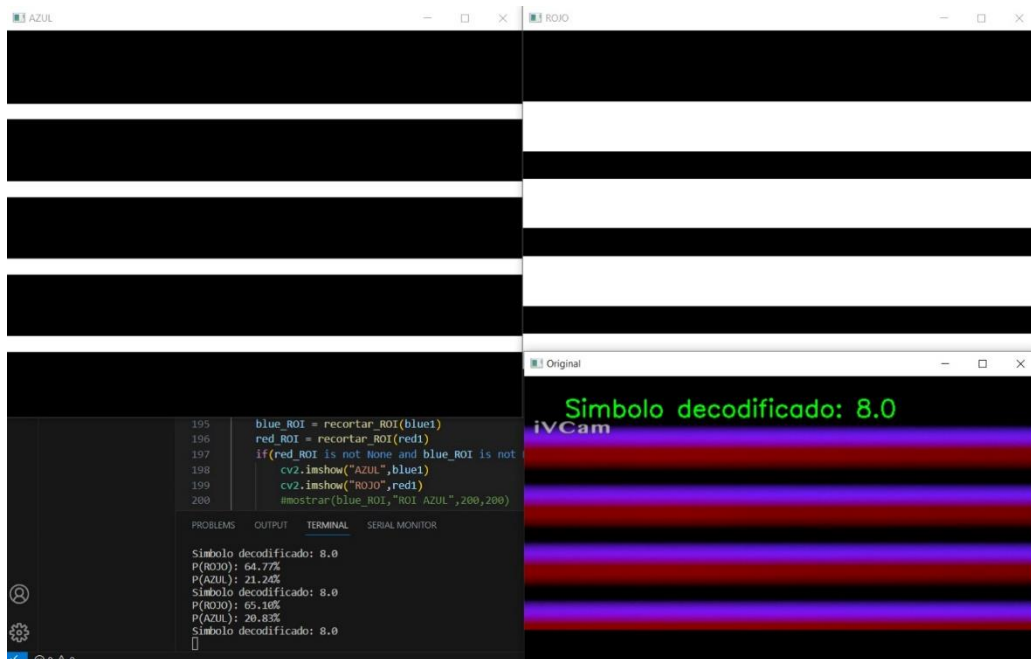
Decodificación del "7"



Nota. En la imagen se muestra el número 7 procesado y decodificado.

Figura 4646

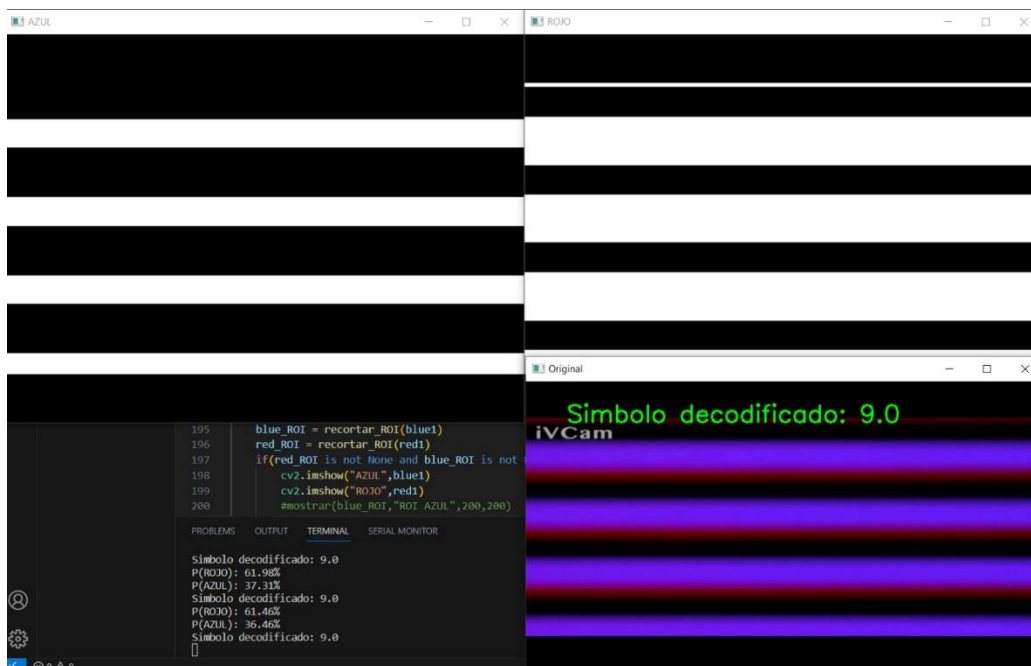
Decodificación del "8"



Nota. En la imagen se muestra el número 8 procesado y decodificado.

Figura 4747

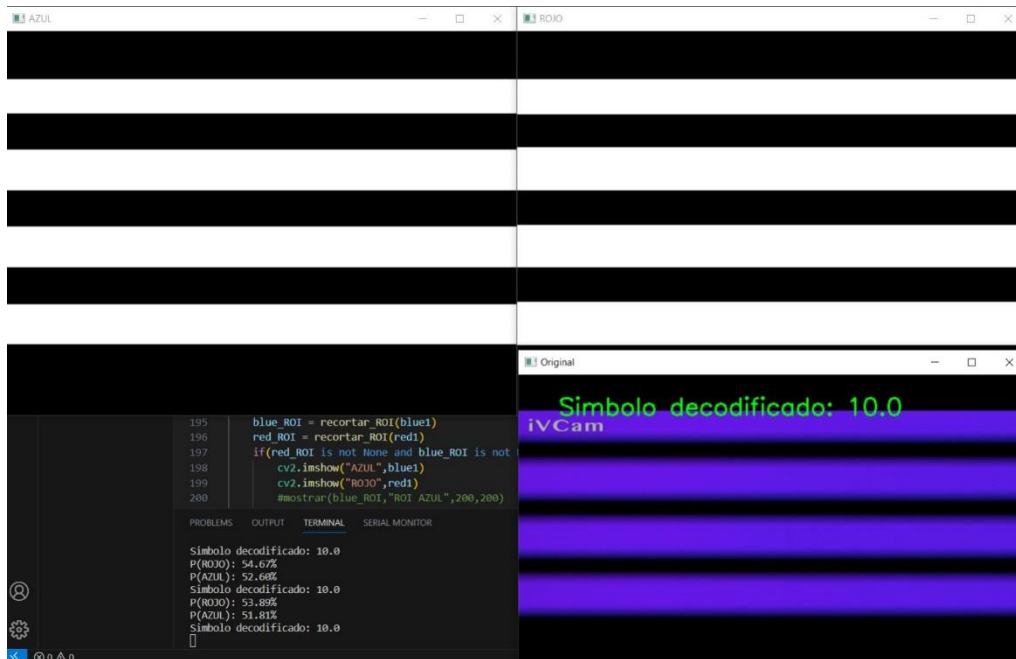
Decodificación del "9"



Nota. En la imagen se muestra el número 9 procesado y decodificado.

Figura 4848

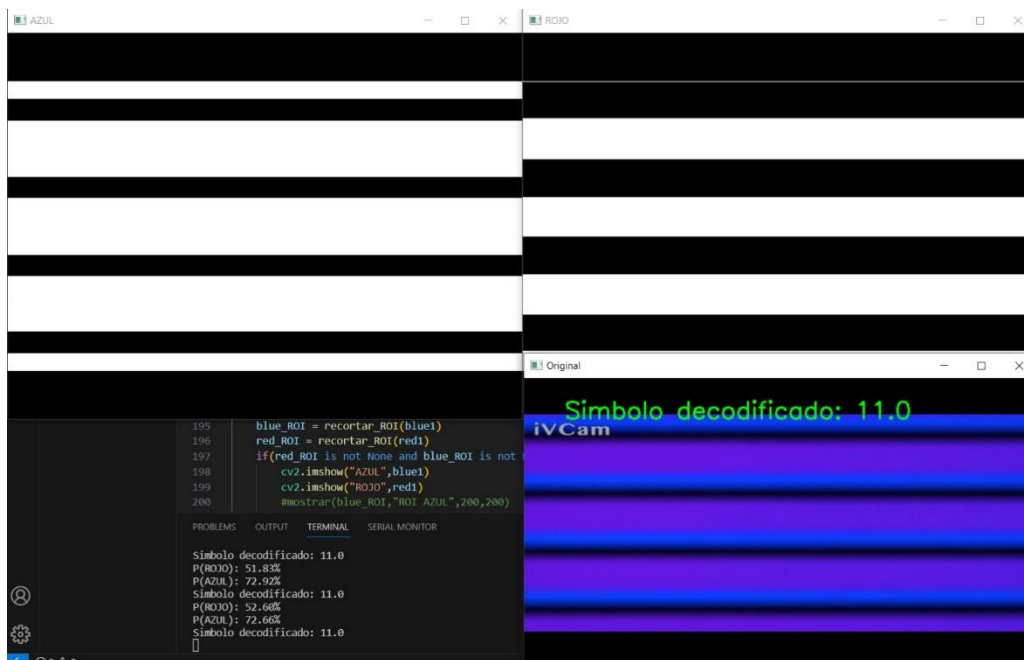
Decodificación del "10"



Nota. En la imagen se muestra el número 10 procesado y decodificado.

Figura 4949

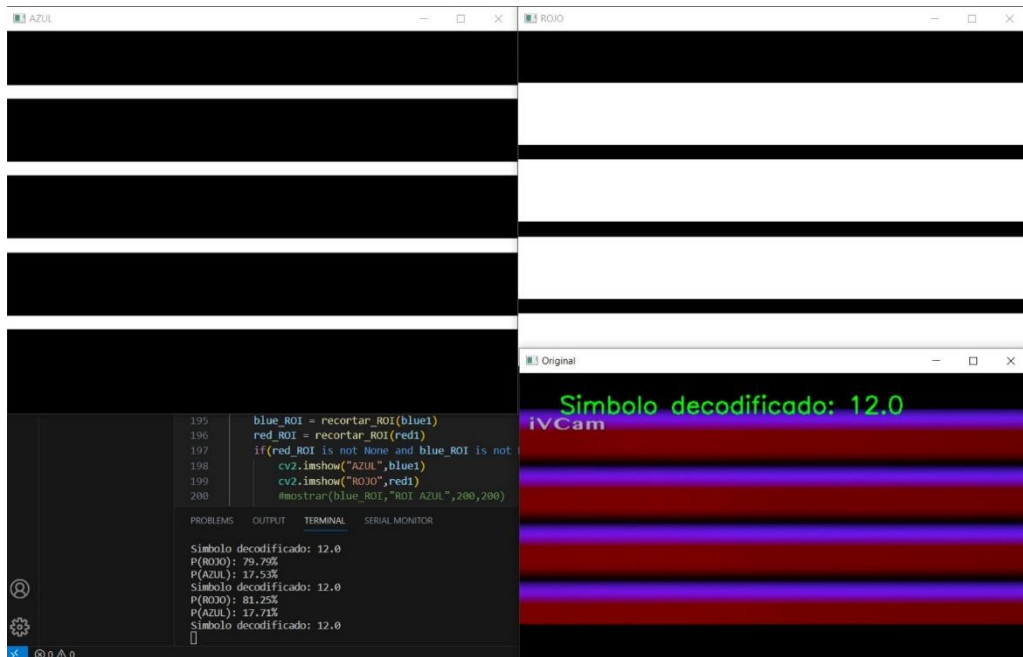
Decodificación del "11"



Nota. En la imagen se muestra el número 11 procesado y decodificado.

Figura 5050

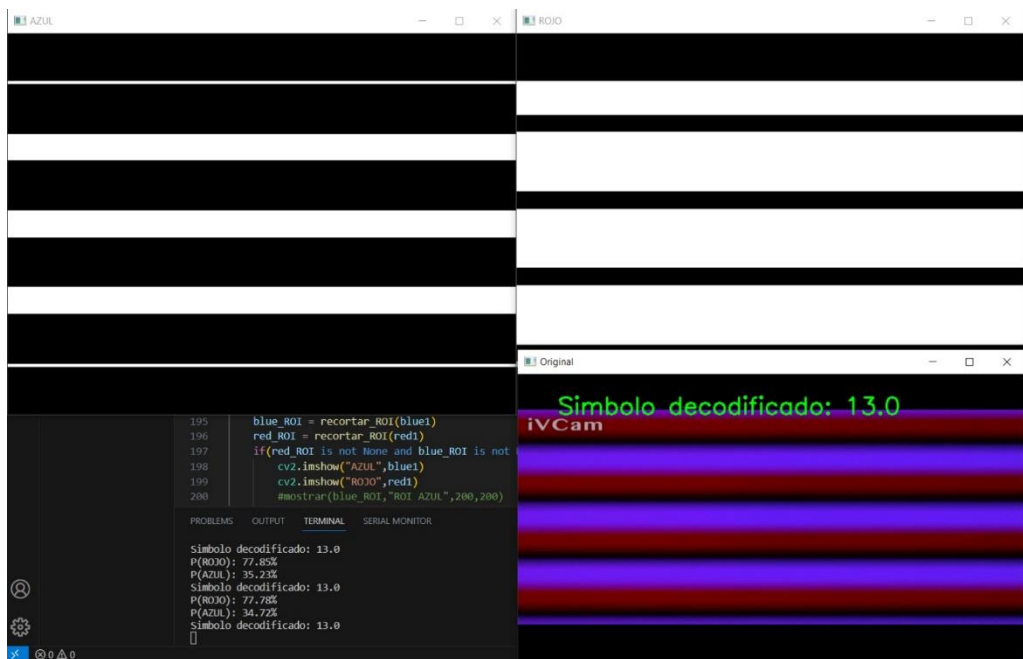
Decodificación del "12"



Nota. En la imagen se muestra el número 12 procesado y decodificado.

Figura 5151

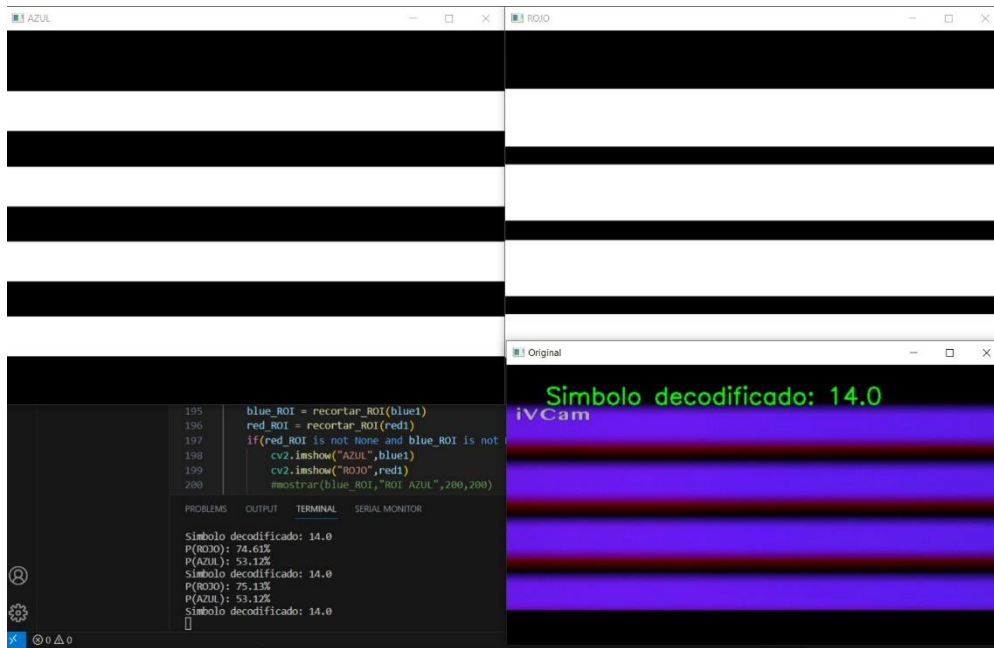
Decodificación del "13"



Nota. En la imagen se muestra el número 13 procesado y decodificado.

Figura 5252

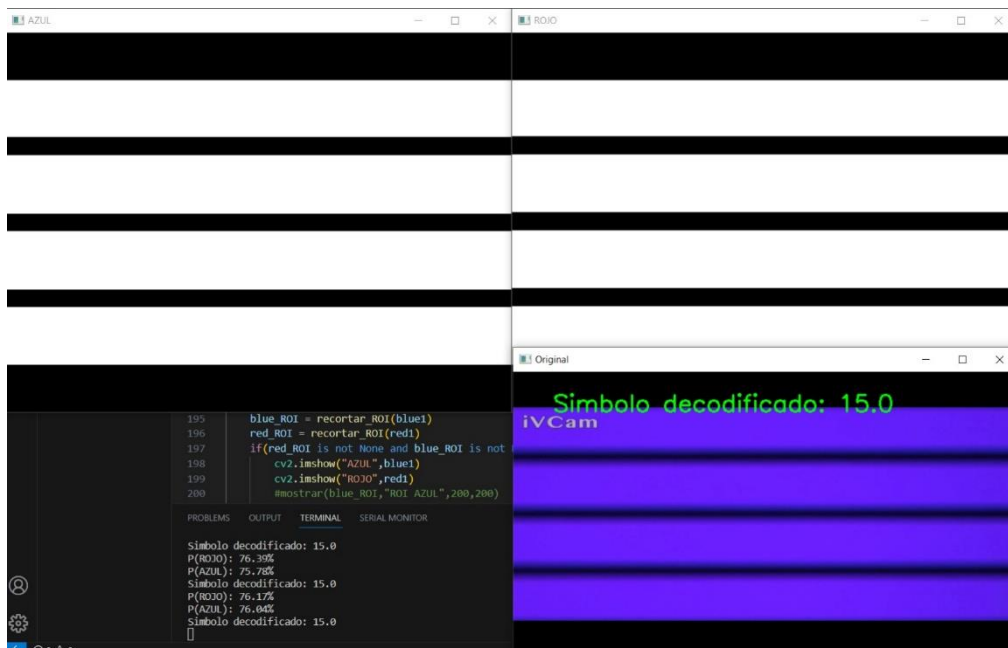
Decodificación del "14"



Nota. En la imagen se muestra el número 14 procesado y decodificado.

Figura 5353

Decodificación del "15"



Nota. En la imagen se muestra el número 15 procesado y decodificado.

Anexo C. Datos enviados en las pruebas de transmisión y recepción

Tabla 2

Números enviados.

Número enviado	Número recibido	Estado
Primer Conjunto		
<i>0</i>	<i>0</i>	<i>Decodificado</i>
<i>1</i>	<i>1</i>	<i>Decodificado</i>
<i>2</i>	<i>2</i>	<i>Decodificado</i>
<i>3</i>	<i>3</i>	<i>Decodificado</i>
<i>4</i>	<i>4</i>	<i>Decodificado</i>
<i>5</i>	<i>5</i>	<i>Decodificado</i>
<i>6</i>	<i>6</i>	<i>Decodificado</i>
<i>7</i>	<i>7</i>	<i>Decodificado</i>
<i>8</i>	<i>8</i>	<i>Decodificado</i>
<i>9</i>	<i>9</i>	<i>Decodificado</i>
Segundo Conjunto		
<i>0</i>	<i>1</i>	<i>No Decodificado</i>
<i>1</i>	<i>1</i>	<i>Decodificado</i>
<i>2</i>	<i>2</i>	<i>Decodificado</i>
<i>3</i>	<i>3</i>	<i>Decodificado</i>
<i>4</i>	<i>4</i>	<i>Decodificado</i>
<i>5</i>	<i>5</i>	<i>Decodificado</i>
<i>6</i>	<i>6</i>	<i>Decodificado</i>
<i>7</i>	<i>7</i>	<i>Decodificado</i>
<i>8</i>	<i>8</i>	<i>Decodificado</i>
<i>9</i>	<i>9</i>	<i>Decodificado</i>

Tercer Conjunto		
0	1	<i>No Decodificado</i>
1	1	<i>Decodificado</i>
2	2	<i>Decodificado</i>
3	3	<i>Decodificado</i>
4	4	<i>Decodificado</i>
5	5	<i>Decodificado</i>
6	6	<i>Decodificado</i>
7	7	<i>Decodificado</i>
8	8	<i>Decodificado</i>
9	9	<i>Decodificado</i>
Cuarto Conjunto		
0	0	<i>Decodificado</i>
1	1	<i>Decodificado</i>
2	2	<i>Decodificado</i>
3	3	<i>Decodificado</i>
4	4	<i>Decodificado</i>
5	5	<i>Decodificado</i>
6	6	<i>Decodificado</i>
7	7	<i>Decodificado</i>
8	8	<i>Decodificado</i>
9	9	<i>Decodificado</i>
Quinto Conjunto		
0	0	<i>Decodificado</i>
1	1	<i>Decodificado</i>
2	2	<i>Decodificado</i>
3	3	<i>Decodificado</i>

4	4	<i>Decodificado</i>
5	5	<i>Decodificado</i>
6	6	<i>Decodificado</i>
7	7	<i>Decodificado</i>
8	8	<i>Decodificado</i>
9	9	<i>Decodificado</i>

Nota. Datos enviados en las pruebas de transmisión y recepción.

REFERENCIAS

- Rajan Sagotra, R. A. (2013). Visible Light Communication. *International Journal of Computer Trends and Technology (IJCTT)*, 5. Obtenido de <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=ca7d4a34e2efaa7f1cb0fd569baf43dec3cef080>
- C. Q. Sánchez. (2013). Transmisión de datos por medio de sistemas VLC. Vector Plus. chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://accedacris.ulpgc.es/bitstream/10553/11881/1/0231633_00038_0005.pdf
- M. Z. Chowdhury, M. T. Hossan, A. Islam and Y. M. Jang. (2018). A Comparative Survey of Optical Wireless Technologies: Architectures and Applications. *IEEE Access*, vol. 6, pp. 9819-9840. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8259465>
- N. Saeed, S. Guo, K. H. Park, T.Y. Al-Naffouri, M.S. Alouini. (2019). Optical camera communications: Survey, use cases, challenges, and future trends. *Physical Communication*, vol. 37. <https://www.sciencedirect.com/science/article/abs/pii/S1874490719304859>
- Izquierdo, A. M. (2021, diciembre 3). La importancia de la tecnología en la educación actual. *Luca*. <https://www.sciencedirect.com/science/article/abs/pii/S1874490719304859>
- Los efectos de la tecnología en el trabajo. (2023, marzo 13). Randstad. <https://www.randstad.es/tendencias360/como-influye-la-tecnologia-en-el-entorno-laboral/>
- Francesco Vasca, L. L. (2012). *Dynamics and Control of Switched Electronic Systems*. Obtenido de https://link.springer.com/chapter/10.1007/978-1-4471-2885-4_2

CristoJV. (2020) Explain like I'm five: Optical Camera Communications (part 1). (s/f). CristoJV.

<https://cristojv.com/explain-like-im-five-optical-camera-communications-part-1.html>

Fluke. (2016, octubre 31). What is duty cycle? Fluke.com. [https://www.fluke.com/en-](https://www.fluke.com/en-us/learn/blog/electrical/what-is-duty-cycle)

[us/learn/blog/electrical/what-is-duty-cycle](https://www.fluke.com/en-us/learn/blog/electrical/what-is-duty-cycle)

OpenCV: Image thresholding (s. f.).

https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html

OpenCV: K-Means Clustering in OpenCV. (s. f.).

https://docs.opencv.org/3.4/d1/d5c/tutorial_py_kmeans_opencv.html

OpenCV: Contours : Getting started. (s. f.).

https://docs.opencv.org/4.x/d4/d73/tutorial_py_contours_begin.html

OpenCV: Contours : Getting started. (s. f.-b).

https://docs.opencv.org/4.x/d4/d73/tutorial_py_contours_begin.html

APÉNDICE

Transmisor

```
const int redPin = 19;
const int bluePin = 5;
const int freq = 240;
const int redChannel = 0;
const int blueChannel = 2;
const int resolution = 10;

int dutyCycle = 0;
int dutyRed = 0;
int dutyBlue = 0;

void codificarPWM(int number){
  int cont1 = 0;
  for(int i=0; i<4; i++){
    for(int j=0; j<4; j++){
      if(cont1 == number){
        int tmpRed = 20*(i+1);
        int tmpBlue = 20*(j+1);
        dutyRed = ((pow(2, resolution) - 1)*tmpRed)/100;
        dutyBlue = ((pow(2, resolution) - 1)*tmpBlue)/100;
        Serial.print("Se codificó el simbolo: ");
        Serial.println(number);
        Serial.print("R: ");
        Serial.print(tmpRed);
        Serial.print("% ");
        Serial.println(dutyRed);
        Serial.print("B: ");
        Serial.print(tmpBlue);
        Serial.print("% ");
        Serial.println(dutyBlue);
        return;
      }
      cont1 = cont1 + 1;
    }
  }
  Serial.println("No se puede codificar simbolo");
}

void setup(){

  Serial.begin(9600);
```

```
    ledcSetup(redChannel, freq, resolution);

    ledcSetup(blueChannel, freq, resolution);

    ledcAttachPin(redPin, redChannel);

    ledcAttachPin(bluePin, blueChannel);

}

void loop(){

    if (Serial.available() > 0) {
        int numeroRecibido = Serial.parseInt();

        while (Serial.read() != '\n');

        if (numeroRecibido >= 0 && numeroRecibido <= 15) {

            Serial.print("Número recibido: ");
            Serial.println(numeroRecibido);
            codificarPWM(numeroRecibido);
            ledcWrite(redChannel, dutyRed);
            ledcWrite(blueChannel, dutyBlue);
        } else {
            Serial.println("Error: Número fuera del rango (0-15).");
        }
    }
}
```


Receptor

```
import cv2
import numpy as np
import time
import datetime
from pushbullet import Pushbullet

#-----+
#   VARIABLES           |
#-----+
# Indice de la camara, generalmente la webcam integrada es 0 y la USB adicional es 1
# Si solo tienes una camara puedes dejarlo en 0
camera_index = 0

#-----+
#   FUNCIONES           |
#-----+
def porcentaje_valores_pixeles(imagen_binaria):
    # Contar la cantidad de pixeles con valor 0 (fondo) y 255 (objeto)
    total_pixeles = imagen_binaria.shape[0] * imagen_binaria.shape[1] #total de pixeles (total de filas
    *total de columnas)
    pixeles_fondo = total_pixeles - cv2.countNonZero(imagen_binaria) #cantidad pixeles de fondo (se
    resta los pixeles que no son 0)
    pixeles_objeto = cv2.countNonZero(imagen_binaria) #los pixeles que no son 0 son los objeto

    # Calcular los porcentajesq
    porcentaje_fondo = (pixeles_fondo / total_pixeles) * 100
    porcentaje_objeto = (pixeles_objeto / total_pixeles) * 100

    return porcentaje_fondo, porcentaje_objeto

def compensar_valor(porcentaje):
    valor_compensado = -1
    if(porcentaje>=10 and porcentaje<30):
        return 20
    if(porcentaje>=30 and porcentaje<50):
        return 40
    if(porcentaje>=50 and porcentaje<70):
        return 60
    if(porcentaje>=70 and porcentaje<90):
        return 80
    return valor_compensado
```

```
def decodificar_simbolo(valor_red, valor_blue):
```

```
    s = -1
    valor_red = compensar_valor(valor_red)
    valor_blue = compensar_valor(valor_blue)
```

```
    if(valor_red>0 and valor_blue>0):
        s = (valor_red/5) + (valor_blue/20) - 5
    return s
```

```
def k_mean_cluster(frame1):
```

```
    floatFrame = np.float32(frame1) #crea un frame con pixeles de 32 bits y ya no de 8
    clusters = (cv2.TERM_CRITERIA_MAX_ITER + cv2.TERM_CRITERIA_MAX_ITER, 10, 1.0)
    pixels = 2
    _,label,center =
cv2.kmeans(floatFrame,pixels,None,clusters,10,cv2.KMEANS_RANDOM_CENTERS) #necesita
que el frame sea de 32 bits
    center = np.uint8(center)
    res = center[label.flatten()]
    res2 = res.reshape((floatFrame.shape))
    return res2
```

```
def filtrar(frame, limite: int):
```

```
    _,binary = cv2.threshold(frame,limite,255,cv2.THRESH_BINARY)
    binaryk = k_mean_cluster(binary)
    _,filtered_frame = cv2.threshold(binaryk,127,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)
    return filtered_frame
```

```
def recortar_ROI(imagen_binaria, area_minima=100):
```

```
    # Encontrar los límites superior, inferior, izquierdo y derecho de las franjas blancas
    contornos,_ = cv2.findContours(imagen_binaria, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
```

```
    # Filtrar contornos pequeños (área menor a 'area_minima')
    contornos_filtrados = [contorno for contorno in contornos if cv2.contourArea(contorno) >=
area_minima]
```

```
    if not contornos_filtrados:
        return None
```

```
    min_y = imagen_binaria.shape[0]
    max_y = 0
    min_x = imagen_binaria.shape[1]
```

```

max_x = 0

for contorno in contornos_filtrados:
    (x, y, w, h) = cv2.boundingRect(contorno)
    min_y = min(min_y, y)
    max_y = max(max_y, y + h)
    min_x = min(min_x, x)
    max_x = max(max_x, x + w)

# Ordenar los contornos de arriba hacia abajo según su posición en el eje y
contornos = sorted(contornos_filtrados, key=lambda c: cv2.boundingRect(c)[1])

# Obtener el índice de inicio de la segunda franja y el índice de inicio de la última franja
if len(contornos) < 3:
    return None

segundo_inicio_y = cv2.boundingRect(contornos[1])[1]
ultimo_inicio_y = cv2.boundingRect(contornos[-1])[1]
# Recortar la imagen original utilizando los límites encontrados
imagen_recortada = imagen_binaria[segundo_inicio_y:ultimo_inicio_y, min_x:max_x]

if(imagen_recortada.shape[0]==0 or imagen_recortada.shape[1]==0):
    return None

return imagen_recortada
#-----

#-----MAIN ACTIVITY#-----#

# Crear objeto cap para la captura de video
cap = cv2.VideoCapture(camera_index)
# Verificar si la cámara se abrió correctamente
if not cap.isOpened():
    print("No se pudo abrir la cámara")
    exit()

tiempo_anterior = time.time()

simbolo = -1
while (True):

    # Capturar frame por frame

```

```

ret, frame = cap.read()

frame640x360 = cv2.resize(frame,(640,360))
font = cv2.FONT_HERSHEY_SIMPLEX
if(simbolo == -1):
    # describe the type of font
    cv2.putText(frame640x360,f"No se detectan datos",(50, 50),font, 1,(0, 255, 0),2,cv2.LINE_4)
else:
    cv2.putText(frame640x360,f"Simbolo decodificado: {simbolo}",(50, 50),font, 1,(0, 255,
0),2,cv2.LINE_4)

cv2.imshow('Original', frame640x360)
# Si el frame se leyó correctamente ret es True
if not ret:
    print("No se puede recibir el frame (stream end?). Exiting ...")
    break

# divide el frame por canal BGR
blue,green,red = cv2.split(frame)
#Filtrar imagen
blue1 = filtrar(blue, 180)
red1 = filtrar(red, 60)
#Recortar region de Interes
blue_ROI = recortar_ROI(blue1)
red_ROI = recortar_ROI(red1)
if(red_ROI is not None and blue_ROI is not None):
    cv2.imshow("AZUL",blue_ROI)
    cv2.imshow("ROJO",red_ROI)
else:
    cv2.imshow("AZUL",blue1)
    cv2.imshow("ROJO",red1)
tiempo_actual = time.time()
if (tiempo_actual - tiempo_anterior > 3):
    tiempo_anterior = time.time()
    if ((red_ROI is not None) and (blue_ROI is not None)):
        # Llamar a la función para obtener los porcentajes
        _, porcentaje_red = porcentaje_valores_pixeles(red_ROI)
        _, porcentaje_blue = porcentaje_valores_pixeles(blue_ROI)
        # Mostrar los resultados
        print(f"P(ROJO): {porcentaje_red:.2f}%")
        print(f"P(AZUL): {porcentaje_blue:.2f}%")
        #Decodificar Simbolo
        simbolo = decodificar_simbolo(porcentaje_red, porcentaje_blue)

print(f"Simbolo decodificado: {simbolo}")

```

```
pb = Pushbullet("o.puC314nDvizphTKdTGG63Z3tnybnS9kJ")
```

```
push = pb.push_note("NÚMERO DECODIFICADO", str(simbolo))  
else:
```

```
print(f"No se detecta ningun símbolo")
```

```
pb = Pushbullet("o.puC314nDvizphTKdTGG63Z3tnybnS9kJ")
```

```
push = pb.push_note("NÚMERO DECODIFICADO", "No se detecta ningun símbolo")
```

```
# Salir del bucle si se presiona 'q'
```

```
if cv2.waitKey(1) == ord('q'):
```

```
    break
```

```
# Cuando todo está hecho, libera la captura de video
```

```
cap.release()
```

```
cv2.destroyAllWindows()
```