



Comunicación MATLAB - Arduino: Abordando Blink y conceptos fundamentales

MATLAB - Arduino communication: Addressing Blink and fundamental concepts

Lisbette Priscila Rodriguez Villacis^{1*}

¹Universidad Estatal Península de Santa Elena. Facultad de Sistemas Y Telecomunicaciones.
Santa Elena - La Libertad. Ecuador

ORCID: <https://orcid.org/0000-0002-5333-5950>

Correo: priscila25rodriguez@gmail.com

Raúl Jair Villon Tomala²

²Universidad Estatal Península de Santa Elena. Facultad de Sistemas Y Telecomunicaciones
Santa Elena - La Libertad. Ecuador

ORCID: <https://orcid.org/0000-0002-5658-2877>

Correo: rvillon05@gmail.com

*Autor para correspondencia: priscila25rodriguez@gmail.com

Resumen

La comunicación entre MATLAB y Arduino se han convertido en herramientas esenciales, su desarrollo en la electrónica y la programación. Se desglosan conceptos fundamentales detrás de esta investigación. Se detallarán guías de ejemplos prácticos mediante (“Blink”), como la posibilidad de encender y apagar un diodo emisor de luz, la intención es proporcionar una orientación en cuanto a configuración de pines digitales, operaciones. Se abordan ejemplos donde podremos ver funcionamiento de servomotores, con lector de pines analógicos y digitales para tener información contextual. Antes de dar paso a comunicación de mayor entendimiento, se debe entender el comando por serial para poder enviarlos a Matlab, siendo este un conocimiento técnico valioso. Estudiaremos la lectura del (IMU) Unidad de Medición Inercial para capturar datos y usarlos para crear experiencias visuales en movimiento, mediante esta práctica se aprenderá a fusionar datos obtenido de la IMU con las capacidades Matlab y generar visualización tridimensional. Los lectores tendrán una comprensión sólida de cómo hacer la programación y controlarla, a través de Matlab, colocando las bases para futuros proyectos más complejos en el área de automatización y robótica.

Palabras claves: MATLAB; Blink; medición; electrónica

Abstract

Communication between MATLAB and Arduino have become essential tools, their development in electronics and programming. Fundamental concepts behind this research are broken down. Practical example guides will be detailed using (“Blink”), such as the possibility of turning a light-emitting diode on and off, the intention is to provide guidance regarding digital pin configuration and operations. Examples are addressed where we can see the operation of servomotors, with analog and digital pin readers to have contextual information. Before giving way to communication of greater understanding, the serial command must be understood to be able to send them to MATLAB, this being valuable technical knowledge. We will study the reading of the Inertial





Measurement Unit (IMU) to capture data and use it to create visual experiences in motion, through this practice we will learn to merge data obtained from the IMU with MATLAB capabilities and generate three-dimensional visualization. Readers will have a solid understanding of how to do programming and control it, through MATLAB, laying the foundation for future more complex projects in automation and robotics.

Keywords: MATLAB; Blink; measurement; electronics

I. Introducción

El objetivo principal de la comunicación de Matlab-Arduino utilizando la herramienta de visualización Blink, esta función nos permite apreciar los datos recolectados de ciertos dispositivos tanto digitales como analógicos. La automatización y la electrónica en la actualidad han llegado a encontrarse presentes en todas las áreas de nuestra vida moderna, lo que ha llevado a tener un desarrollo considerable de manera significativa fortalece las empresas industriales, universitarias. [1]

Matlab-Arduino consiste en un programa que es cargado en un microcontrolador por medio del entorno que nos ayuda a desarrollar Arduino, y este es el que recibe instrucciones mediante puerto serial, va ejecutando comandos suele devolver resultados y de esta forma es que se realiza la interfaz. Otro Apartado que nos ayuda a estudiar el Arduino como una tarjeta para adquirir Datos es que se encuentra corriendo con el mismo programa, tiene una frecuencia de muestreo lo que puede ser ya sea constante como independiente de lo que estamos simulando, Matlab le ayuda indicando comandos. [2]

La guía tiene por finalidad instruir la convergencia de ambos entornos, se inicia la implementación de un ejercicio a manera inicial, eficaz para manipular un diodo de emisor de Luz. Si se te orientas de manera detallada a la configuración de la comunicación entre los programas, podrás descubrir cómo usarlos para generar el movimiento y control con precisión.

Matlab es un software que es orientado a cálculos matemáticos. Lo que hace adecuado a este programa es el procesamiento de gráficos. En su lugar Arduino es basado en hardware y software de fácil uso, puede ayudar desde el procesamiento de imágenes para obtener los datos de una computadora mediante Matlab. [3]

La lectura de los pines digitales y analógicos de Arduino, a través de Matlab, el propósito de guiarlos de manera sencilla permitiéndote expandir conocimiento en esta interacción creativa.

La tarjeta Arduino es una placa compatible a Matlab, lo que nos permite la fácil adquisición de datos. Matlab en su lugar ofrece paquete Arduino IO que obtiene conexión entre sí. Este envía datos digitales a los pines de Arduino, tomando en cuenta el comando reconocido, Arduino recibe los datos digitales, mostrando salidas en leds de acuerdo con su comando. [4]

Unidad de medición Inercial implementamos una guía respecto a este dispositivo electrónico, Nos indica que registra información obtenida mediante velocidades, orientaciones, se encarga de conseguir lecturas precisas es ideal para trabajar con placa Arduino. Se sabe que es un dispositivo formado por acelerómetro y giroscopio, capaz de mesurar fuerza, pero no ángulos. La IMU da los valores en enteros de 16 bits, Arduino guarda en menos bits por lo que se declaran sus variables y estas almacenaran enteros provenientes de la IMU como tipo enteros especiales. Tenemos `int16_t AcX, AcY, AcZ` son los `raw_values` es decir valores en bruto. [5]





II. Materiales y métodos

Esta guía de investigación de carácter descriptivo y experimental. Dentro de esta búsqueda de retención de información empleamos las bases de datos, Scholar, Scielo, Web of Science, entre otras. Las palabras claves han sido seleccionadas refiriéndonos a comunicación entre Matlab y Arduino y adquisiciones de datos entre sí, para así tener una búsqueda más selectiva y poder entender los conceptos fundamentales, los pasos para la comprensión de este utilizando el Arduino y el programa Matlab sus términos empleados fueron: Comunicación serial, Encender y apagar leds, INU junto con ambas bases, entradas analógicas y digitales, aplicación de servomotores en Matlab-Arduino. Para la búsqueda de conceptos fundamentales y experimentales, se han dado según sus años de publicación durante los 2013 a 2018. Las tesis y artículos elegidos han sido analizados de manera selectiva con el contenido necesario. Durante la recolección de información se tiene en cuenta las tipologías como: artículos a texto entero, casos experimentales del tema. Criterios de exclusión: perspectivas, guías, blogs, resúmenes. Este trabajo cumplirá con normas y principios establecidos en las organizaciones internacionales del campo llamado bioética, es decir no involucrarse con proyectos en los cuales la información es difusa, y usada con fines deshonestos, así como resguardar la integridad de los autores y su propiedad intelectual, realizando un correcto citado bajo las Normas Vancouver.

A. Componentes

Para esta investigación, se emplean elementos físicos y lógicos. Entre ellos el uso de placa Arduino uno, el sensor ultrasónico HC-SR04, servo motor, junto con el software Matlab, que nos permite hacer las practicas. [6]

B. Primeros pasos para usar el programa Blink

Se utilizan una placa Arduino, el software Matlab y leds. Es necesario primero establecer una comunicación entre MATLAB/ARDUINO para empezar a trabajar. Conecta Arduino al ordenador y abre Matlab. Debemos iniciar la conexión por USB con la siguiente línea de código en Matlab (en el comand windows).

arduino = arduino('/dev/ttyS101')

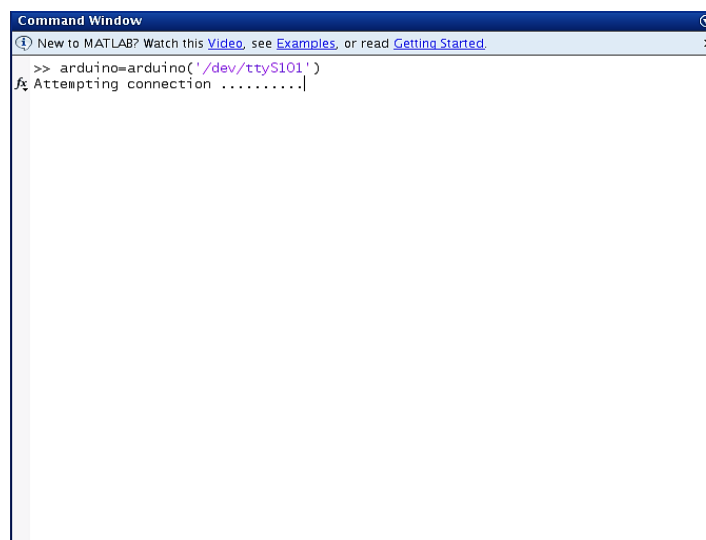


Figura 1. Conexión MATLAB y Arduino





Ahora para programar un encendido y apagado de LED, se crea un bucle mediante las líneas de código. Un ejemplo que se puede demostrar es que al utilizar lo siguiente. Realiza la activación y desactivación del LED en tiempos donde siguen la curva sinusoidal hasta 90° , equivalente a $\frac{\pi}{2}$ que se mide en radianes, que utiliza MATLAB.

C. Control de servomotores

Control de servomotores mediante la sinergia de estas herramientas poderosas. Para aquellos que se encuentran en etapas iniciales de exploración, es importante establecer una comunicación fluida entre Arduino y MATLAB, ya que esto resulta fundamental para facilitar la interacción y llevar a cabo con éxito este tutorial ejemplar. A lo largo de esta guía, abarcaremos el amplio campo de los servomotores, desde los conceptos fundamentales hasta aplicaciones de mayor complejidad, aprovechando la sinergia existente entre las capacidades de MATLAB y Arduino.

Creamos un simple bucle para mover el servomotor de 0 a 180 grados y al revés. Recuerda que puedes detener el programa con “Ctrl+C” en cualquier momento, sin embargo, es recomendable esperar a que el programa finalice por sí solo. Luego establecemos las condiciones de nuestro servo mediante la creación de variables de posición, de igual manera podemos establecer el tiempo en el que le tomara llegar a la posición indicada por el usuario.

D. Comandos por serial

Los Materiales para usar son los Arduino Uno, cable de impresora y ordenador MATLAB, el código es básico, consta de dos variables “i” y “k”. La primera variable aumentara hasta llegar a 30, después se vuelve a 0. La segunda variable va a mantenerse a 12. Cuando cargamos el código en el Arduino, nos dirigimos a abrir Matlab. Y lo primero que y anotamos el siguiente comando. Se crea un bucle “i” y “k” en dos listas. Se crea un bucle for con variable c.

E. Lectura de una IMU por serial

El enfoque en la “MPU-6050” te permitirá experimentar con los datos de aceleración y giroscopio para transformarlos en movimiento virtual del cubo en la pantalla. Utilizamos material como MPU 6050 un Ordenador con Matlab, cable compatible para Arduino. Utiliza el IDE de Arduino para escribir un programa que lea los datos de la IMU y los transmita a través del puerto serie (Serial). En Matlab, puedes utilizar la función serial para establecer una conexión serial con Arduino y recibir los datos de la IMU. En el bucle while, puedes procesar y analizar los datos de la IMU según tus necesidades.

F. Trazado de las lecturas del sensor “DHT11” en Matlab.

Los Materiales que se usan son Arduino uno, software Matlab y sensor de temperatura DHT11. Empezamos con la compilación del código en Arduino usando Arduino IDE. Paso 1 incluya la biblioteca para el sensor “DHT11” que es “DHT.h”. Luego defina el pin de datos “DHT11” conectado al Arduino. Tomamos una función llamada “bucle vacío”, configuras el valor de temperatura y humedad como tipo flotador. Describimos en Arduino:





G. Fabricación de radar usando Arduino

Escribe un programa en el IDE de Arduino que configure el sensor ultrasónico y envíe datos de distancia al puerto serie. Utiliza Matlab para leer los datos de distancia que envía el Arduino a través del puerto serie. Puedes usar la función serial en Matlab para configurar la comunicación serie con el Arduino.

Usando un servomotor, sensor ultrasónico y un Arduino uno, desarrollamos conectando el sensor ultrasónico (5V y GND) para Arduino (5V y GND). El sensor ultrasónico (SIG) en el Arduino va (pin digital 8). Se conecta ahora un motor servo (5V y GND) al Arduino (5V y GND), luego el motor servo (SIG) va en el Arduino pin 9. Para nuestro código en MATLAB: se modificará la línea 28 (`s1=serial('COM12')`) ; que la nueva COM puerto #. Cerrar el puerto serie escribiendo "`s1.close()`" en el “workspace de MATLAB”. Puedes procesar y visualizar los datos de distancia recibidos del Arduino. Puedes crear un gráfico en tiempo real que muestre los objetos detectados y su distancia.

III. Resultados

Análisis e interpretación: Los resultados obtenidos muestran que se efectúa el encendido y apagado del LED en intervalos de tiempo que va siguiendo una función seno hasta poder alcanzar los 90°, este es equivalente a $\frac{\pi}{2}$ si se mide en radianes ya que es la unidad que nos muestra y utiliza Matlab.

Análisis e interpretación: Arduino transmite un valor numérico como dato serial, es interpretado y registrado por Matlab. Interactúa como puerto serial (Matlab), como si se tratase de un archivo de texto que capture valores y analiza con facilidad. El resultado será como la figura 2, en el que se muestra el recorrido de la variable “i” y “k”.

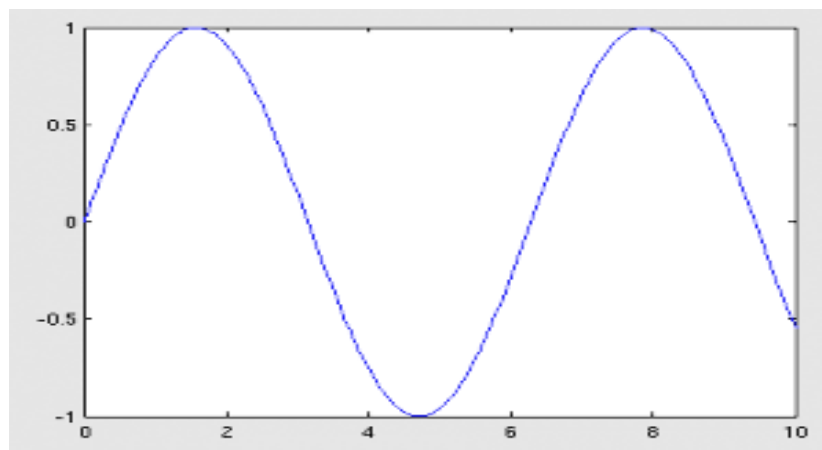


Figura 2. Gráfica del control del encendido y apagado del led

Fuente: Raúl Villon.

Análisis e interpretación: La visualización de datos en la gráfica revela patrones y tendencias que podrían pasar desapercibidos en presentaciones numéricas tendremos las lecturas del sensor. Cuando el código se ejecuta la ventana abrirá un gráfico en tiempo real. Los datos se actualizarán cada dos segundos, ya que este es el tiempo que tarda el sensor “DHT11” en que envíe los datos al puerto serial.



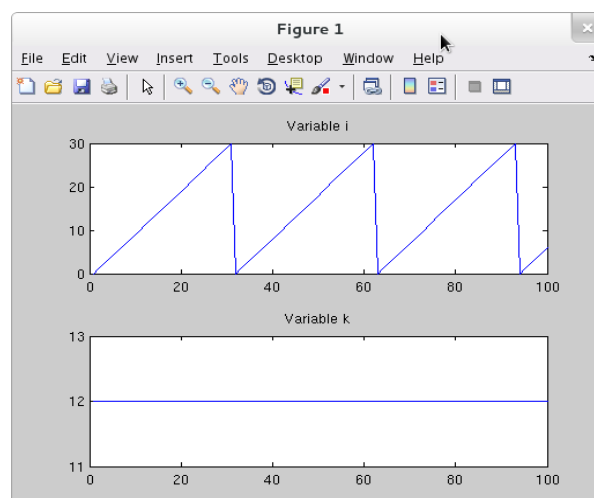


Figura 3: Gráfico del recorrido de las variables "k" e "i"

Análisis e interpretación: Obtendremos una visión práctica de como integrar sensores ultrasónicos y motores servo para desarrollar un radar básico, a continuación, la gráfica del funcionamiento. En este instante el radar sencillo usa un ultrasónico y servo para poder graficar su información.

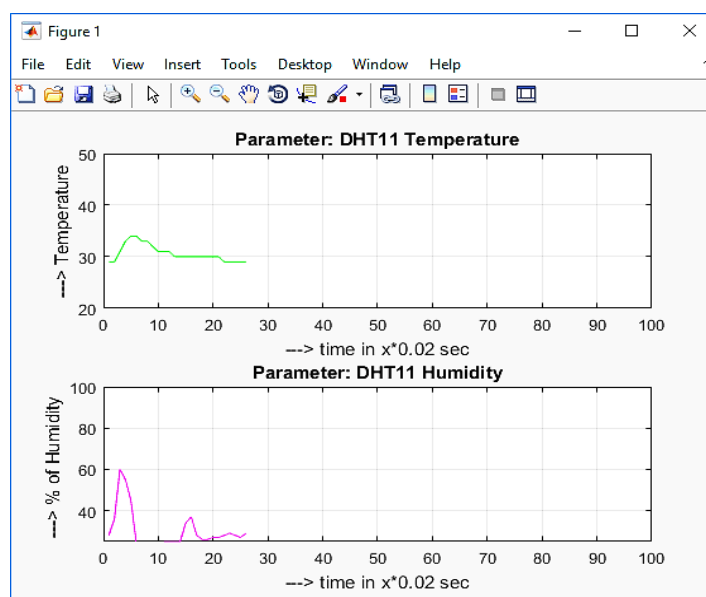


Figura 4. Gráfica de los parámetros de temperatura
Fuente: Raúl Villon

Diferentes tipos de señales digitales y analógicas, los parámetros a enfocar a continuación según la realización.



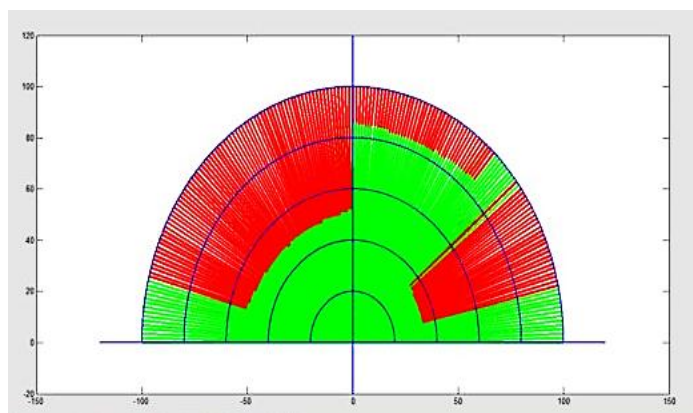


Figura 5. Gráfica del funcionamiento del radar

Fuente: Raúl Villon

Tabla 1. Resumen pruebas

Tipo Señal	Sensor	MF (Grados)	
Análoga	presión	En este caso se toma los valores de la señal de presión y se aplica.	✓
	Temperatura DHT-11	En este caso los datos que se toma de la temperatura se los aplica en diferentes técnicas de señal por medio de MATLAB	✓
Pulsos Digitales			✓
Digital estados	Presencia o Movimiento	Se obtienen datos del un ultrasónico de presencia o movimiento, dentro del rango que se este previsto.	

Fuente: Basantes Hugo

En Matlab se envían datos mediante puertos serie, con el objetivo de mantener facilidad para cada experimentación (control e identificación), teniendo en cuenta que los datos se mostraran de manera que se observara una forma sencilla a través del monitor serie de Arduino. [7]

Base de datos utilizando funciones, data base toolbox conocida como VQB, interfaz fácil de usar para recuperar datos. Los recupera en matriz de células Matlab para poder utilizar el conjunto de funciones. Es importante conocer valores digitales y analógicos, dentro de este programa lo único q vamos a usar es la lectura de valores analógicos, en el pin que deseamos escoger, y tras haber concluido las lecturas, mandaremos datos al Puerto Serial. [8]

El programa Matlab consta con diferentes simuladores de ondas que tienen propiedades, características y al obtener datos de los sensores. Se analizan aquellos datos para su garantía y puede servir para diferentes proyectos. [9]

Se encuentra un uso en la enseñanza, adquisición de datos, etc. La sencillez de uso de esta plataforma llamada





Arduino puede hacer que las personas que no sean expertas en electrónica usen dispositivos, de manera accesible. Como se ve, existen aplicaciones que utilizan la plataforma Arduino como un dispositivo de entrada y salida que envía y lee datos, eso nos lleva a realizar monitorización. En este periodo la plataforma ha ido evolucionando con el tiempo, y el crecimiento de número de usuarios es considerable; también se ha ampliado programas que puedan comunicarse con la plataforma y expanden su utilización. [10]

Los ejemplos demostraron su desempeño consistente. Sus datos fueron precisos y estables lo que ayudó a que se mantengan en línea con valores ya previstos. Queremos decir que estos sistemas son capaces de procesar datos y adquirirlos de manera fiable, obteniendo resultado de sus sensores. La prueba que demuestran los sistemas, de capacidad oportuna. [11]

La presente investigación tiene con alto impacto la enseñanza de programas como MATLAB y Arduino, por su dinámica estructural a nivel superior. En los ejemplos ya dados, se requirieron conexiones en los dispositivos electrónicos fueron capaces de dar como resultado las adquisiciones de datos. Así mismo el manejo fácil de adaptar para adquirir señales de un mismo modo el manejo digital de servomotores. [12]

Se utiliza procesamiento de señales para procedimientos de visión artificial, control, robótica y otros campos. Puede que los análisis de datos mayores se ejecuten, Se puede integrar más lenguajes que nos permita implementar más algoritmos. [13]

IV. Conclusiones

La comunicación, entre nuestros dos programas representa un puente, entre el mundo del software y el hardware, les permite a los desarrolladores o programadores crear soluciones avanzadas que incluye datos y control de dispositivos. Se puede ver una conexión bidireccional. La capacidad que tiene Matlab y Arduino para que se convierta en un recurso para desarrollar sistemas de control.

La clave para dominar la comunicación y comprensión de estas herramientas es practicar hasta adaptarse con los protocolos de comunicación. La flexibilidad y su potencial de expansión hace que la tecnología sea viable para cualquier programador. El uso de este artículo realizado por estudiantes puede ser compartido con las demás personas que deseen entrar una vez más al mundo de MATLAB con Arduino, se necesita paciencia y atención.

Referencias

- [1] O. G. González., *Diseño e implementación de Prácticas de laboratorio para asignaturas de la carrera de ingeniería automática en base al Arduino Uno y Matlab*, Santiago de Cuba, 2016.
- [2] M. D. L. H. KÖLLMER, *Sistemas de Adquisición de Datos basados en la plataforma Arduino. Aplicaciones a Matlab, simulink y android*, 2013.
- [3] G. C. Perfecto, *TRANSMISIÓN DE IMÁGENES DE MATLAB A ARDUINO VÍA PUERTO SERIA*, 2016.
- [4] K. B. A. DIAZ y N. J. C. CASTILLO, *IMPLEMENTACIÓN DE UN MÓDULO DE RECONOCIMIENTO DE VOZ PARA NIÑOS MEDIANTE PROCESAMIENTO DE SEÑALES APLICADO EN UN CASO PRACTICO*, Riobamba , 2017.
- [5] E. M. Merino, *DISEÑO Y CONTROL DE UN SISTEMA ELECTROMECHANICO INESTABLE DE DOS GRADOS DE LIBERTAD: APLICACION AL CASO DE UN PENDULO INVERTIDO*, Barcelona, 2015.
- [6] L. E. U. MEJÍA, *DISEÑO E IMPLEMENTACIÓN DE UN ROBOT HEXÁPODO CONTROLADO*





INALÁMBRICAMENTE MEDIANTE ARDUINO Y ANDROID EQUIPADO CON UN DIRECTOR ELECTRO ÓPTICO., Guayaquil, 2017.

- [7] R. L. Martín, *Adaptación de dispositivo basado en Arduino para pruebas de control por computador*, Sevilla, 2017.
- [8] M. P. Esteso, *MATLAB + Arduino: Serial Port Communication*, 2015.
- [9] F. E. C. ILLESCAS, “*Desarrollo de una Mesa Vibratoria para Simulación de Sismos en estructuras*”, 2022.
- [10] J. C. H. Herranz y J. S. Allende, *UNA MIRADA AL MUNDO ARDUINO*, MADRID, 2015.
- [11] H. G. B. Patiño, *SISTEMA DE ACOPLAMIENTO FÍSICO DE SEÑALES DE SENSORES A SISTEMAS EMBEBIDOS*, IBARRA, 2023.
- [12] C. G. Filiberto, I. R. Mora, I. V. d. l. Cruz, F. Méndez y A. L. R. Carlos, *Interfaz Arduino-Simulink para la medición en paralelo de dos sensores MPU6050*, MICHOACÁN, 2020.
- [13] V. M. R. Z. G. C. G. Gonzales Carrasco Matías Sebastián, Chiclayo, Lambayeque, 2022.
- [14] C. A. S. C. A. D. C. S. B. J. C. P. S. R. O. M. Alejandro Jose Briones Romero, *Diseño de un relé de sobrecorriente con característica de tiempo inverso usando Arduino y Matlab.*, Ecuador, 2022.
- [15] M. D. L. H. KÖLLMER, *Sistemas de Adquisición de Datos basados en la plataforma Arduino. Aplicaciones a Matlab, Simulink y Android.*, Leganes, 2013.

