



UNL

Universidad
Nacional
de Loja

Universidad Nacional de Loja

Facultad de la Energía las Industrias y Recursos Naturales no Renovables

Carrera de Ingeniería Electromecánica

Diseño y construcción de un prototipo de persiana con ajuste automático de iluminación para las aulas de bloque 3 de la **FEIRNNR**

Trabajo de Titulación, previo a la
obtención del título de Ingeniero
Electromecánico

AUTOR:

Jimmy Paúl Carreño Ordóñez

DIRECTOR:

Ing. Fernando V. Ramírez Cabrera M. Sc.

Loja – Ecuador

2023

Certificación

Loja, 12 de septiembre de 2023

Ing. Fernando V. Ramírez Cabrera M.Sc.

DIRECTOR DEL TRABAJO DE TITULACIÓN

CERTIFICO:

Que he revisado y orientado todo el proceso de elaboración del Trabajo de Titulación denominado: **Diseño y construcción de un prototipo de persiana con ajuste automático de iluminación para las aulas del bloque 3 de la FEIRNNR**, previo a la obtención del título de **Ingeniero Electromecánico**, de la autoría del señor estudiante **Jimmy Paúl Carreño Ordóñez**, con **cédula de identidad Nro. 1105228728**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja, para el efecto, autorizo la presentación del mismo para su respectiva sustentación y defensa.

Ing. Fernando V. Ramírez Cabrera M.Sc.

DIRECTOR DEL TRABAJO DE TITULACIÓN

Autoría

Yo, **Jimmy Paúl Carreño Ordóñez**, declaro ser autor del presente Trabajo de Titulación y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos, de posibles reclamos y acciones legales, por el contenido del mismo. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja la publicación de mi Trabajo de Integración Curricular o de Titulación, en el Repositorio Digital Institucional – Biblioteca Virtual.

Firma:



Cédula de identidad: 1105228728

Fecha: 23 de noviembre del 2023

Correo electrónico: jpcarreno@unl.edu.ec

Teléfono: 0991098246

Carta de autorización por parte del autor, para consulta, reproducción parcial o total y/o publicación electrónica del texto completo, del Trabajo de Titulación.

Yo, **Jimmy Paúl Carreño Ordóñez**, declaro ser autor del Trabajo de Titulación denominado: **Diseño y construcción de un prototipo de persiana con ajuste automático de iluminación para las aulas del bloque 3 de la FEIRNNR**, como requisito para optar por el título de **Ingeniero Electromecánico**, autorizo al sistema Bibliotecario de la Universidad Nacional de Loja para que, con fines académicos, muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido en el Repositorio Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional, en las redes de información del país y del exterior con las cuales tenga convenio la Universidad. La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del Trabajo de Titulación que realice un tercero.

Para constancia de esta autorización, suscribo, en la ciudad de Loja, a los veintitrés días del mes de noviembre de dos mil veintitrés.

Firma:

Autor: Jimmy Paúl Carreño Ordóñez

Cédula de identidad: 1105228728

Dirección: Barrio Bolonia, Av. Villonaco

Correo electrónico: jpcarreenoo@unl.edu.ec

Teléfono: 0991098246

DATOS COMPLEMENTARIOS:

Director del Trabajo de Titulación: Ing. Fernando V. Ramírez Cabrera M.Sc.

Dedicatoria

Este triunfo se lo dedico a mi madre Martha Ordóñez, por siempre brindarme su apoyo incondicional a lo largo de toda mi carrera universitaria y haber confiado en mis capacidades dándome la fortaleza día a día para no darme por vencido, a mi padre Rodrigo Carreño (+) por los valores que me enseñó en vida y por siempre brindarme ánimos para seguir adelante, a todos mis hermanos; y de manera especial a mi Esposa Erika Yauri y a mi hijo Matías Sebastián, que han sabido estar a mi lado en todo momento y gracias a su comprensión y apoyo incondicional estoy culminando mi ansiada carrera.

Jimmy Paúl Carreño Ordóñez

Agradecimiento

Quiero agradecer de manera especial, a Dios, por darme la sabiduría y fortaleza para no darme por vencido y poder cumplir una de mis metas, a la Universidad Nacional de Loja por abrirme sus puertas para estudiar en esta gran institución, a la carrera de Ingeniería electromecánica por la infinidad de conocimientos compartidos a lo largo de toda mi formación académica y mi profundo agradecimiento a mi director de tesis, el Ing. Ing. Fernando Cabrera, quien con su sabiduría, paciencia, experiencia y conocimiento me ha sabido orientar de la mejor manera para poder culminar con éxito mi trabajo de titulación. Finalmente, mi gratitud a cada uno de los docentes que formaron parte de mi carrera universitaria.

Jimmy Paúl Carreño Ordóñez

Índice de contenidos

Portada.....	i
Certificación.....	ii
Autoría.....	iii
Carta de autorización.....	iv
Dedicatoria.....	v
Agradecimiento.....	vi
Índice de contenidos.....	vii
Índice de tablas.....	x
Índice de figuras.....	xi
Índice de anexos.....	xii
1. Título.....	1
2. Resume.....	2
Abstract.....	3
3. Introducción.....	4
3.1. Objetivo General.....	5
3.2. Objetivo Específico.....	5
4. Marto Teórico.....	6
4.1. Capítulo I: Parámetros de iluminación.....	6
4.1.1. Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables.....	6
4.1.2. Principales fundamentos de la iluminación.....	6
4.1.3. Geometría Solar.....	10
4.1.4. Niveles de iluminación para aulas.....	10
4.1.5. Depósito de control de iluminación natural.....	11
4.2. Capítulo II: Fundamento electrónico.....	12
4.2.1. Internet de las cosas (IoT).....	12
4.2.2. Módulo sensor de luz BH1750.....	14
4.2.3. EasyDriver A3967.....	14
4.2.4. Motor paso a paso.....	16
4.2.5. Sistema de control lazo abierto.....	18
4.2.6. Integrated Development Environment (IDE).....	19
4.2.7. Módulo ESP32.....	20

4.2.8. Comunicación Inter Integrated Circuit (I2C).....	21
4.3. Capítulo III: Android Studio.....	22
4.3.1. Estructura de proyectos.....	22
4.3.2. Herramientas de depuración y perfil.....	23
4.3.3. Entorno de desarrollo de Android.....	24
4.3.4. Lenguaje de programación Java.....	25
4.3.5. JavaScript Object Notation (JSON).....	26
4.3.6. Firebase.....	27
4.3.7. Realtime Database.....	27
5. Metodología.....	29
5.1. Área de estudio.....	29
5.2. Equipos y materiales.....	29
5.2.1. Equipos.....	29
5.2.2. Materiales.....	30
5.3. Procedimientos.....	30
5.3.1. Primer objetivo: Identificar los niveles de iluminación requeridos para el edificio 3 de la Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables.....	30
5.3.2. Segundo objetivo: Diseñar un prototipo de persianas automáticas que regulen el nivel de iluminación en las aulas.....	32
5.3.3. Tercer objetivo: Construir el prototipo diseñado para validar su correcto funcionamiento.....	35
5.4. Procesamiento.....	36
5.4.1. Primer objetivo: Identificar los niveles de iluminación requeridos para el edificio 3 de la Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables.....	36
5.4.2. Segundo objetivo: Diseñar un prototipo de persianas automáticas que regulen el nivel de iluminación en las aulas.....	36
5.4.3. Tercer objetivo: Construir el prototipo diseñado para validar su correcto funcionamiento.....	37
6. Resultados.....	38
6.1. Levantamiento de la información.....	38
6.1.1. Mallado de la zona de emplazamiento.....	38

6.1.2. Niveles de iluminación.....	39
6.2. Diseño de prototipo.....	40
6.2.1. Creación de un proyecto en Firebase.....	41
6.2.2. Programación de la tarjeta ESP32.....	41
6.2.3. Diseño de aplicación en Android Studio.....	44
6.2.4. Funcionamiento integral del prototipo.....	45
6.3. Implementación del prototipo.....	46
6.3.1. Construcción del prototipo.....	46
6.3.2. Implementación del prototipo en la zona de emplazamiento.....	48
6.3.3. Calibración.....	50
7. Discusión.....	53
8. Conclusiones.....	54
9. Recomendaciones.....	55
10. Bibliografía.....	56
11. Anexos.....	60
11.1. Anexo 1. Práctica Experimental.....	60
11.2. Anexo 2. Planos de investigación.....	75
11.3. Anexo 3. Algoritmos de programación en Arduino.....	81
11.4. Anexo 4. Algoritmo de programación en Java: Android Studio.....	86
11.5. Anexo 5. Datasheet del sensor BH1750.....	91
11.6. Anexo 6. Datasheet del 17HS4401.....	113
11.7. Anexo 7. Certificado de traducción del resumen.....	115
11.8. Anexo 8. Productos disponibles en el mercado.....	117

Índice de tablas

Tabla 1. Valores de iluminación mínima, para aulas de centros educativos.....	11
Tabla 2. Configuraciones del resolución del sensor.....	15
Tabla 3. Materiales requeridos para el prototipo	30
Tabla 4. Nivel de iluminación en la zona de emplazamiento.....	40
Tabla 5. Variables especificadas en la base de datos.....	42
Tabla 6. Materiales usados para la construcción del prototipo.....	46
Tabla 7. Características de motores de persianas disponibles en el mercado.....	50

Índice de tablas

Figura 1. Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables.....	6
Figura 2. El espectro electromagnético.....	7
Figura 3. Persiana de láminas vertical.....	12
Figura 4. Arquitectura IoT en cinco etapas.....	13
Figura 5. EasyDiver A3967.....	15
Figura 6. Motores paso a paso.....	17
Figura 7. Momento torsor de un disco.....	18
Figura 8. Sistema de control de lazo abierto.....	18
Figura 9. Entorno de trabajo IDE.....	20
Figura 10. Tarjeta ESP32.....	21
Figura 11. Archivos de proyectos en la vista de proyectos de Android.....	23
Figura 12. Interfaz Andriod Studio.....	25
Figura 13. Ubicación geográfica del lugar de estudio.....	29
Figura 14. Flujograma de metodología del primer objetivo.....	31
Figura 15. Flujograma de metodología del segundo objetivo.....	32
Figura 16. Flujograma de programación del prototipo.....	33
Figura 17. Librería a implementar para la tarjeta.....	34
Figura 18. Flujograma de programación del prototipo.....	36
Figura 19. Mallado del área de estudio.....	38
Figura 20. Niveles de iluminación del Aula A-3-2-2.....	39
Figura 21. Medidas de la ventana del aula A-3-2-2.....	40
Figura 22. Base de Datos creada en Firebase.....	42
Figura 23. Interfaz gráfica de la aplicación móvil.....	44
Figura 24. Comunicación integral del prototipo.....	45
Figura 25. Prueba del funcionamiento del prototipo.....	45
Figura 26. Materiales que conforman el control del prototipo.....	47
Figura 27. Soldadura del circuito.....	47
Figura 28. Ensamble del prototipo.....	48
Figura 29. Implementación del prototipo.....	48
Figura 30. Dimensiones y peso de una sección de las aletas de las persiana.....	49
Figura 31. Mecanismos de movimiento de la persiana vertical.....	49
Figura 32. Instalación y prueba de funcionamiento del prototipo.....	51

Figura 33. Niveles de medición obtenidos para calibración.....	52
Figura 34. Productos disponibles en el mercado.....	117
Figura 35. Controlador domótico para persiana vertical.....	118

Índice de anexos

Anexo 1. Práctica Experimental.....	60
Anexo 2. Planos de investigación.....	75
Anexo 3. Algoritmos de programación en Arduino.....	81
Anexo 4. Algoritmo de programación en Java: Android Studio.....	86
Anexo 5. Datasheet del sensor BH1750.....	91
Anexo 6. Datasheet del 17HS4401.....	113
Anexo 7. Certificado de traducción del resumen.....	115
Anexo 8. Productos disponibles en el mercado.....	117

1. Título

Diseño y construcción de un prototipo de persiana con ajuste automático de iluminación para las aulas del bloque 3 de la FEIRNNR

2. Resumen

La luz es una onda electromagnética que estimula la retina del ojo humano para percibir el color; al ser este recurso altamente eficiente con respecto al rendimiento de color. Por lo tanto, la Universidad Nacional de Loja al promover la innovación para resolución de problemáticas sociales, en la presente investigación se implementó un prototipo de persianas automáticas que permitan regular el nivel de luminosidad en las aulas de FEIRNNR; la cual presenta tecnología IoT enfocada a mejorar la ergonomía de los estudiantes.

Para ello, se calculó los promedios de iluminación en el interior y exterior del aula, dando como resultado 1531,21 lx en la parte interior y 43540.6464 lx de iluminación en la parte exterior, obteniendo un factor de luz diurna del 3.56 % y niveles de iluminación superiores a lo exigido por las normativas. Por lo cual, se diseñó una aplicación móvil denominada PerIoT capaz de controlar el movimiento de una persiana vía web, de forma manual o automática, de la misma forma, se lo puede realizar de forma física desde el panel de control diseñado; permitiendo tener una mayor versatilidad con respecto a otros mecanismos de control, de la misma forma, su costo hace que el prototipo se competitivo, ya que en el mercado local no existen sistemas de control de ese tipo. Además, el prototipo de manera automática permite aprovechar los niveles de luz natural que existe en el exterior, siendo este un preámbulo hacia la eficiencia energética dentro del aula.

Palabras clave: IoT, persiana, Firebase, Android Studio, iluminación.

Abstract

Light is an electromagnetic wave that stimulates the retina of the human eye to perceive color; as this resource is highly efficient with respect to color rendering. Therefore, the Universidad Nacional de Loja by promoting innovation to solve social problems, in this research a prototype of automatic blinds was implemented to regulate the level of brightness in the classrooms of FEIRNNR; which presents IoT technology focused on improving the student ergonomics. For this purpose, the lighting averages inside and outside the classroom were calculated, resulting in 1531.21 lx in the interior part and 43540.6464 lx of lighting in the exterior part, obtaining a daylight factor of 3.56% and lighting levels higher than those required by the regulations. Therefore, a mobile application called PerIoT was designed, capable of controlling the movement of a blind via web, manually or automatically, in the same way, it can be done physically from the control panel designed; allowing greater versatility with respect to other control mechanisms, in the same way, its cost makes the prototype competitive, since in the local market there aren't control systems of this type. In addition, the prototype automatically allows to take advantage of the levels of natural light that exists outside, this being a preamble to energy efficiency in the classroom.

Keywords: IoT, rolling shutter, Firebase, Android Studio, lighting.

3. Introducción

La luz es una onda electromagnética de diversas longitudes, que estimula la retina del ojo humano para percibir el color. De la misma forma, la luz natural es una fuente muy eficiente al momento de proporcionar un rendimiento de los colores; por lo cual, su aprovechamiento y control ha sido objeto de múltiples estudios. En este caso se proporciona el prototipo de un dispositivo de control de iluminación natural, para un aula de la FEIRNNR basado en tecnología IoT; con el cual se hace una introducción a nuevas tecnologías que transmiten información vía web orientado hacia la eficiencia energética. De la misma forma, en la actualidad no existen investigaciones relevantes con respecto al control automático de la iluminación natural, originando un vacío de conocimiento en esta parte de gestión energética, que busca adaptar modelos que mejoren la ergonomía de las personas y los requerimientos de los usuarios para cumplir con sus actividades.

La Universidad Nacional de Loja, está comprometida con la innovación y búsqueda de nuevas tecnologías enfocadas en solucionar problemáticas sociales. Por ejemplo, Sánchez Cueva (2022) en su investigación de automatización del sistema de iluminación LED para el edificio 3 de la Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables, destaca la necesidad de implementación de un sistema de iluminación que cumpla con los valores de la norma UNE 12464-1, sin embargo, el autor no consideró el control del sistema de iluminación natural. Por otro lado, Silveira Amaro (2020) en su investigación planteó un prototipo de persianas automatizadas mediante microcontroladores para espacios residenciales, en el cual destaca la versatilidad, proyección y facilidad del desarrollo de este tipo de tecnología, para ello hizo uso de la tarjeta ESP32 para obtener un resultado satisfactorio a bajo costo.

En la presente investigación, se realizó la respectiva revisión bibliográfica que define los elementos que conforman el prototipo, así como las diferentes plataformas en donde se desarrollará la interfaz y código fuente del prototipo. Por lo tanto, la metodología de investigación empieza por la identificación de los niveles de iluminación natural que existe en el salón de clases, seguidamente se declara las variables en Firebase y se integra la base de datos de a un proyecto de Android Studio, finalmente, se desarrolla el código de programación en la tarjeta ES32, a la cual se le ensamblan los respectivos actuadores.

La zona de emplazamiento cuenta con un promedio de Factor de Luz Diurna (FLD) de 3.56 %, estando el mismo dentro de los límites de la normalidad y de la misma forma, las mediciones realizadas en el aula A-3-2-2 determinaron niveles de iluminación superiores de los 500 lx que exigen las normativas. Por otro lado, se diseñó un aplicativo móvil denominado PerIoT, con el cual se puede controlar de manera normal o automática el paso de luz natural

de una ventana; de la misma forma, señala el nivel de iluminación externo. Seguidamente, se construyó un prototipo usando como elemento principal la tarjeta ESP32 y se vinculó la información de la misma con una base de datos en Firebase, la cual es modificada por la aplicación móvil.

Por lo tanto, en la presente investigación se implementó un prototipo de persianas automáticas que permitan regular el nivel de luminosidad en las aulas de FEIRNNR, para ello se hizo uso de tecnología IoT y lectura de datos en una base de datos de la red, al dispositivo se lo puede controlar mediante una aplicación móvil a través de internet y mediante un panel de control físico; además, descartando el costo de la persiana, el precio del mecanismo de control es de \$78.30, siendo este competitivo con otros existentes en el mercado internacional que no cuentan con las mismas características que el prototipo propuesto.

3.1 Objetivo general

Implementar un prototipo de persianas automáticas que permitan regular el nivel de luminosidad en las aulas de FEIRNNR

3.2 Objetivos específicos

- Identificar los niveles de iluminación requeridos para el edificio 3 de la Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables.
- Diseñar un prototipo de persianas automáticas que regulen el nivel de iluminación en las aulas.
- Construir el prototipo diseñado para validar su correcto funcionamiento.

4. Marco Teórico

4.1. Capítulo I: Parámetros de iluminación

4.1.1. Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables

La facultad que se muestra en la figura 1, tiene el objetivo de formar profesionales con bases sólidas y enfoques científicos – humanistas, acorde a la vanguardia tecnológica en los avances científicos. Sus instalaciones constan de 5 bloques principales, en donde el edificio del bloque 3 cuenta con 3 plantas y tiene una altura total de 8.4 m y un área superficial de 209.77 m²; la planta baja cuenta con 3 aulas y un taller de robótica; el primer y segundo piso cuenta con 3 aulas, dos baños y una sala de docentes (Sánchez, 2022).



Figura 1: Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables

4.1.2. Principales fundamentos de la iluminación

La luz es una onda electromagnética de diversas longitudes, que estimula la retina del ojo humano para percibir el color. La región de las ondas se denomina espectro visible y ocupa una banda muy estrecha del espectro como se muestra figura 2. Cuando la luz es separada en sus diversas longitudes de onda componentes es llamada espectro, en donde el ojo humano percibe estas diferentes longitudes de onda como colores (Sylvania, 2021).

4.1.2.1. Temperatura del color

Es la comparación entre el color de una fuente luminosa dentro de un espectro luminoso y la luz que emitiría un cuerpo negro calentado a una temperatura determinada, por esta razón se la expresa en Kelvin, ya que es una medida relativa. La ley de Wien relaciona los

conceptos de longitud de onda y temperatura; por lo tanto, cuanto mayor sea la temperatura de un cuerpo negro, menor será la longitud de onda que emite (Sylvania, 2021).

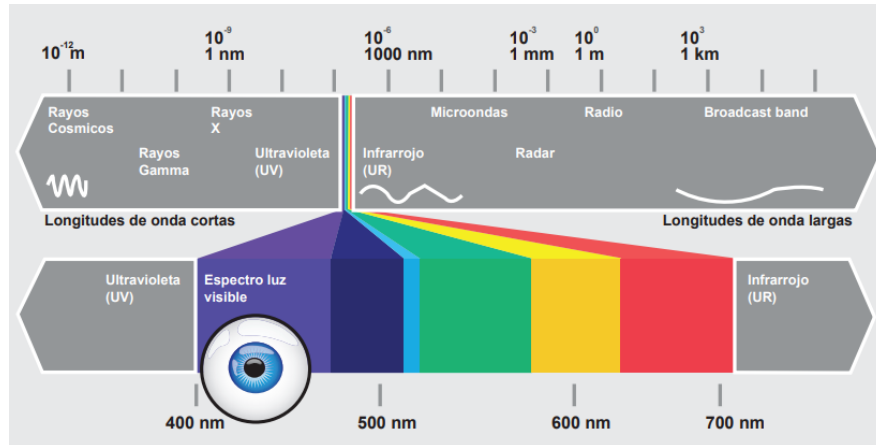


Figura 2: El espectro electromagnético

4.1.2.2. Intensidad Luminosa

Es la energía luminosa emitida en una dirección. Su unidad es la candela (cd), que es una unidad fundamental del S.I., y es aproximadamente la intensidad emitida por una vela. Es decir, es la cantidad de flujo luminoso que emite una fuente por unidad de ángulo sólido (Monroy, 2014).

4.1.2.3. Flujo luminoso

Es la medida de potencia luminosa percibida; que está ajustada para reflejar la sensibilidad del ojo humano a diferentes longitudes de onda. Su unidad de medida es el lumen (lm) y se obtiene de la suma ponderando la potencia para cada longitud de onda del espectro visible, que representa la sensibilidad del ojo en función de la longitud de onda (Sylvania, 2021).

4.1.2.4. Iluminancia

En fotometría, la iluminancia (E) es la cantidad de flujo luminoso que incide sobre una superficie por unidad de área. La unidad de medida tanto de la Emitancia Luminosa como de la Iluminancia en el Sistema Internacional es el lux y se la obtiene de acuerdo a la **Ecuación (1)** (Sylvania, 2021).

$$E = \frac{\phi}{S} \quad \text{Ecuación (1)}$$

Donde:

E: Iluminancia [lx]

S: Superficie de incidencia [m²]

ϕ : Flujo luminoso [lm]

La iluminancia y su distribución en el área de la tarea y en el área circundante tienen un gran impacto en cómo una persona percibe y realiza la tarea visual de un modo rápido, seguro y confortable.

4.1.2.5. Eficiencia Luminosa

Es el flujo emitido por unidad de potencia de las fuentes luminosas (lm/W). El rendimiento de una luminaria permite conocer la cantidad de flujo luminoso aprovechada por la fuente, el cuál es un elemento importante al momento de la selección de una lámpara por el aspecto económico (Monroy, 2014). Se puede calcular mediante la **Ecuación (2)**.

$$R = \frac{lumen}{P} \quad \text{Ecuación (2)}$$

Donde:

P : Potencia consumida por la lámpara [W]

4.1.2.6. Método de análisis estadístico

Calcula para momentos puntuales un contraste con las condiciones del cielo estándar. Las dos variables a considerar son la iluminancia y el Factor de Luz Diurna (FLD) que establece el nivel de iluminación interior con respecto al nivel de iluminación exterior. En la iluminación natural, la dispersión de los rayos luminosos no depende de la separación de la fuente de luz, por lo general solo se trata de luz solar directa y difusa. Por lo general, en un espacio arquitectónico y referencia al objeto de estudio como mesas o escritorios, el plano de análisis se divide mediante una rejilla, calculando la iluminancia en cada una de sus celdas. De esa manera, para cada espacio es posible establecer la iluminancia mínima y máxima, e incluso el porcentaje del área que cuentan con unos determinados valores de iluminancia (Ordóñez, 2021).

De acuerdo con la norma española UNE-EN 12464-1 denominada “Luz e iluminación. Iluminación de los lugares de trabajo Parte 1: Lugares de trabajo en interiores” establece las rejillas para calcular la luminancia, los cuales deben crearse para indicar los puntos en los cuales se calculan y verifican los valores de iluminancia para el área o áreas de la tarea; en donde es preferible utilizar rejillas de celdas cuadradas cuyas dimensiones se la obtiene mediante la **Ecuación (3)** (UNE-12464, 2022).

$$p = 0.2 * 5^{\log_{10}(d)} \quad \text{Ecuación (3)}$$

Donde:

$p < 10$ m

d : Es la dimensión mayor del área [m]

p : Es el tamaño máximo de la celda o rejilla [m]

Por otro lado, el factor de luz diurna (FLD) establece el impacto que la geometría del espacio y la cantidad de acristalamiento tiene en los niveles de iluminación natural. Por lo general se expresa como porcentaje y se define como la relación entre la iluminancia en un punto plano horizontal interior producida por la luz recibida directa o indirectamente; y la iluminancia en un plano exterior producida por un hemisferio no obstruido del mismo cielo, el cual se lo representa mediante la **Ecuación (4)**.

$$FactordeLuzDiurna(FLD) = \frac{E_{int}}{E_{ext}} \quad \text{Ecuación (4)}$$

Donde:

E_{int} : Iluminación producida en un punto del plano de trabajo interior.

E_{ext} : Iluminación producida por un cielo no obstruido en un punto de un plano exterior.

El cálculo del FLD tiene estrecha relación con el punto anterior, solo se considera el flujo luminoso del cielo, no la luz directa del sol. Los métodos de cálculo y aplicación del FLD han evolucionado relativamente poco a lo largo del tiempo, aunque los valores recomendados han variado drásticamente (desde 0.1 % hasta 10 %) dependiendo del tipo de edificio, las expectativas regionales y el periodo histórico (Ordóñez, 2021).

4.1.2.7. Iluminación Natural

La longitud de onda de la luz visible va desde los 380 nm hasta los 780 nm; esto es lo que percibe el ojo humano. La luz natural es una fuente muy eficiente que proporciona un rendimiento de los colores de manera eficiente. La disponibilidad y características de la luz natural dependen de la latitud, meteorología, época del año y del momento del día. Es sabido que la cantidad de luz natural recibida en la tierra varía con la situación, la proximidad a las costas o tierra adentro. La luz natural consta de tres componentes:

- El haz directo procedente del sol
- La luz natural difundida en la atmósfera
- La luz procedente de reflexión

El uso de sistemas de luz natural o de sistemas de control para el alumbrado artificial, influye en el equilibrio energético de los edificios.

4.1.3. Geometría solar

4.1.3.1. Declinación solar

Es el ángulo que forma por el plano que contiene el eje de rotación terrestre y el plano perpendicular a la eclíptica. Sin bien la diferencia absoluta entre el eje de rotación y la perpendicular es de 23.45° . Existen diversas formas de obtener el valor del ángulo de declinación solar. El procedimiento simplificado se expresa mediante la **Ecuación (5)** (Tejeda y Azpeitia, 2015).

$$\delta = \varepsilon_{ob} * \text{sen} \left[\frac{360(284 + n_j)}{365} \right] \quad \text{Ecuación (5)}$$

Donde:

δ : Ángulo de declinación solar en grados.

ε_{ob} : Ángulo de oblicuidad de la eclíptica considerando como constante 23.45°

n_j : Día juliano del año

4.1.3.2. Hora solar efectiva

Tomando en cuenta la latitud de la zona de aplazamiento se puede calcular el intervalo de tiempo solar para cada día del año, para ello se debe hacer uso de la **Ecuación (6)** (Tejeda y Azpeitia, 2015).

$$Wt = \cos^{-1}(-\tan(\text{latitud}) * \tan(\delta)) * \left(\frac{24}{360} \right) \quad \text{Ecuación (6)}$$

Donde:

Latitud: Latitud de la zona de emplazamiento

Para el cálculo del intervalo de tiempo se debe aplicar la **Ecuación (7)** y la **Ecuación (8)**

$$\text{Hora de Salida} = 12 - Wt \quad \text{Ecuación (7)}$$

$$\text{Hora de Puesta} = 12 + Wt \quad \text{Ecuación (8)}$$

4.1.4. Niveles de iluminación para aulas

Los requisitos de iluminación son determinados por la satisfacción de tres necesidades humanas básicas:

- Confort visual; en el que los trabajadores tienen una sensación de bienestar, de un modo indirecto también contribuye a un elevado nivel de la productividad.

- Prestaciones visuales; en el que los trabajadores son capaces de realizar sus tareas visuales, incluso en circunstancias difíciles y durante períodos más largos.
- Seguridad

Por lo tanto, existen algunas normativas que establecen los niveles de iluminación adecuados para un establecimiento, los cuales se describen en la tabla 1.

Tabla 1: Valores de iluminación mínima, para aulas de centros educativos

Normativa	Clase de recinto o actividad	Iluminancia lux	Fuente
DIN 5035	Escuelas o institutos: Salas de conferencias, oficinas, salas de reuniones, bibliotecas, salas de música, aulas, cocinas, galerías de arte, pequeñas salas de enseñanza	250	(Taboada, 1983)
UNE 12464: Luz e iluminación de los lugares de trabajo	Aulas – Actividades generales	500	(UNE-12464, 2022)
NEC. Eficiencia energética en edificaciones residenciales	Cuartos de estudios o trabajo	500	(Norma-Ecuatoriana-de-la-Construcción, 2018)
INEN 1 152. Iluminación natural de edificios	Cuartos de estudios	500	(INEN-1.152, 1984)

Cabe destacar que el porcentaje de factor de luz natural (FLD) de acuerdo a la INEN 1 152 (1984) es del 1.9 % para escuelas, específicamente para salones de clase.

4.1.5. Dispositivos de control de iluminación natural

4.1.3.2. Persianas

Las persianas como se muestra en la figura 3 son un dispositivo mecánico formado por un conjunto de láminas delgadas que pueden movilizarse para permitir el paso de luz natural y para el control de la privacidad, además pueden ubicarse en puertas o ventanas. Las persianas suelen fabricarse en diferentes materiales como tela, PVC, madera, plástico o metal que se ajustan mediante al giro de las piezas (Silveira Amaro, 2020).

Los mecanismos más habituales para mover una persiana son mediante motor eléctrico y a través de una cinta o escalerilla que se recoge en una polea inferior (con un muelle interno) y otra superior, en la cual la fuerza la realiza el peso de la persiana.



Figura 3: Persiana de láminas vertical

Los tipos de persianas que existen en el mercado son de tipo alicantinas, que son más robustas y por lo general están hechos de madera o PVC; las de tipo enrollables, que impide totalmente el paso de la luz y aísla la ventana del exterior; las de tipo Venecia que se caracterizan por tener las láminas dispuestas de forma horizontal y que permite regular el paso de la luz. También se encuentran las persianas de tipo vertical cuya función es similar a la de la cortina y su mecanismo es sencillo, como se muestra en la figura 3.

4.2 Capítulo II: Fundamento electrónico

4.2.1. Internet de las cosas (IoT)

Internet of Thing (IoT) es el conjunto de tecnologías consolidadas que permite que objetos cotidianos puedan comunicarse a través de una red de área personal (PAN), red de área local (LAN) y red de área amplia (WAN); recopilando información que nos permita supervisar el estado y comportamiento de dichos objetos. Todo esto basado en la idea de hacer

la vida más cómoda y poder ignorar cosas simples que con una información de variación de datos (big data o macrodatos) pueden controlarse (Cruz Vega et al., 2015).

Los elementos del ecosistema de IoT deben interconectarse, integrarse, administrarse y ponerse a disposición de las aplicaciones. Los pilares fundamentales de la IoT son los objetos conectados, tecnología de la red, protocolos de comunicación, plataforma IoT para el tratamiento de datos y aplicaciones de usuario. La integración de IoT con la Cloud Computing (computación de la nube) aporta muchas ventajas a las diferentes aplicaciones, sin embargo, debido a la gran cantidad de información que se maneja por cada plataforma, es complejo integrar todas las funciones en un solo aplicativo y la comunicación de la información por banda ancha produce interrupciones y problemas de seguridad en el envío de información (Solano, 2022).

Existen diferentes arquitecturas compuestas de varias capas que describen las etapas del desarrollo IoT; las principales son capa de percepción, capa de red y capa de aplicación, sin embargo a medida que va evolucionando esta tecnología se han ido integrando más estructuras como las que se muestra en la 4, las cuales son:

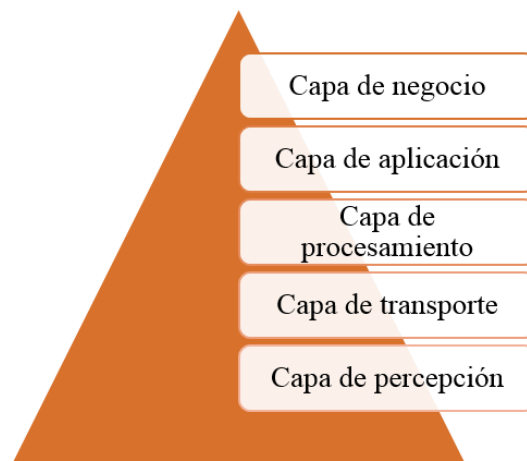


Figura 4: Arquitectura IoT en cinco etapas

Fuente: (Solano, 2022).

- *Capa de recepción:* Se encuentran los sensores y mecanismos de lectura de información.
- *Capa de red:* Conecta a los objetos inteligentes, dispositivos de red y servidores para transmitir y procesar los datos.
- *Capa de aplicación:* Define el lugar en donde se aplicará la IoT y que necesidad va a cubrir.

- *Capa de transporte:* Transfiere los datos mediante redes inalámbricas como wifi, bluetooth, 3g, 5g, Zigbee, etc.
- *Capa de procesamiento o middleware:* Almacena, analiza y procesa grandes cantidades de información, aplicando tecnologías como Big Data.
- *Capa de negocio:* Gestiona el sistema IoT e incluida sus aplicaciones.

4.2.2. Módulo sensor de luz BH1750

Es un sensor que mide la intensidad lumínica y está integrado con un convertidor analógico/digital de 62 bits, dando la información con una señal de salida de forma digital con el protocolo I2C, obteniendo una interacción mucho más sencilla entre sensor y la placa programable, como el Arduino o Raspberry. Algunas de sus principales características son (Pucha, 2023):

- Voltaje de operación de 3 V a 5 V.
- Interfaz digital a través del bus I2C con capacidad de seleccionar entre 2 direcciones.
- Respuesta espectral similar a la del ojo humano.
- Su rango de medición va de 1 a 65535 lx.
- Bajo consumo de energía.
- Rechazo de ruido a 50 o 60 Hz.

Tiene dos direcciones para poder trabajar, en donde el pin ADDR internamente en el módulo tiene una resistencia a GND, por lo que si dejamos sin conectarlo la dirección será 0x23; en donde los módulos de resolución con respecto al tiempo de medición se describen en la tabla 2; tal y como lo describe el datasheet del anexo 5.

4.2.3. EasyDriver A3967

Es un controlador que se utiliza para operar motores paso a paso bipolares en modos completos, medios, cuartos y octavos. El controlador tiene la capacidad de operar con una corriente de salida de 30 V y ± 750 mA. El A3967SLB incluye un regulador de corriente de tiempo fijo que tiene la capacidad de operar en modos de decaimiento de corriente lenta, rápida o mixta. Dentro de las características que presta el control de decaimiento de corriente resulta es una reducción del ruido audible del motor, una mayor precisión del paso y una reducción de la disipación de energía (Mossi, 2016), que se muestra en la figura 34:

Tabla 2: Configuraciones de resolución del sensor

Modo	Resolución	Tipo de medición
High resolution Mode 2	0.5 lx	120 ms
High Resolution Mode	1 lx	120 ms
Low Resolution Mode	4 lx	16 ms

Fuente: (Pucha, 2023)

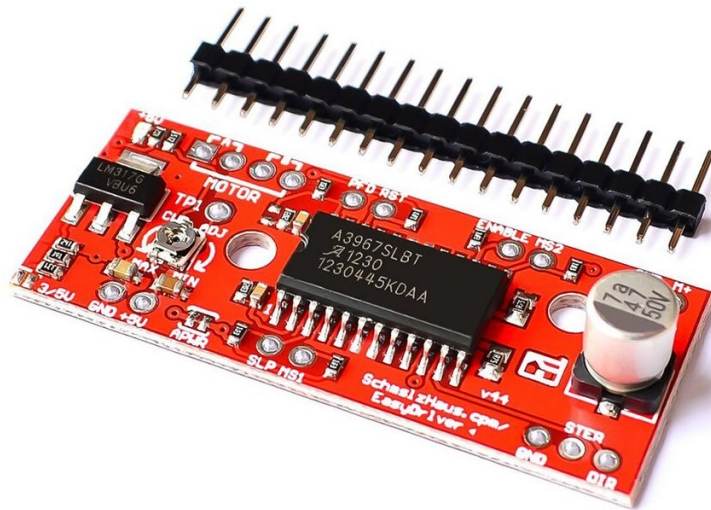


Figura 5: EasyDriver A3967

Este controlador puede proporcionar pulsos de 5V y se puede configurar en 4 posiciones con los pines MS1 y MS2, dentro de las condiciones que presenta este dispositivo como se muestra en el anexo 8, se destaca (Cué, 2018):

- Circuito integrado A3967
- Ajustar la resolución de micro pasos a través de los pins MS1 y MS2 en la placa. Las resoluciones son full, medio, un cuarto y un octavo (por defecto está ajustado en un octavo).
- Compatible con motores paso a paso de 4, 6 y 8 cables de cualquier voltaje.
- Corriente de control ajustable desde 150 mA/fase a 750 mA/fase

- Alimentación desde 5 V a 30 V. El voltaje es directamente proporcional al torque en velocidades altas.

Para el caso de los motores paso a paso, en el entorno de programación IDE de Arduino se debe especificar los pines que señalen el sentido de giro y el número de pulsos a realiza; ya que con sólo dos pines podemos controlar la dirección como el paso (Cámara, 2020), dentro de la descripción de los pines se destaca:

- GND: Es el pin de tierra o masa, el dispositivo cuenta con 3 de ellos.
- M+: Este pin es con el que alimentamos el controlador, acepta una entrada de alimentación de 5 V a 30 V y una corriente de 2 A.
- A (dos pines) y B (dos pines): Estos pines son las conexiones del motor con el controlador: cable azul en el Pin B-, cable verde en el Pin A-, cable amarillo en el Pín B+, cable rojo en el Pin A+.
- STEP (PASO): Este pin es el encargado de producir el movimiento de los motores
- DIR (DIRECCIÓN): Este pin activado a nivel alto o bajo determina la dirección de rotación.
- MS1 / MS2: Pin de entrada digital para controlar la resolución.
- SLP: Modo suspensión, minimiza el consumo de energía

4.2.4. Motor paso a paso

En la actualidad existe una gran variedad de motores empleados en la industria y que varían de acuerdo a su aplicación; por ello los motores paso a paso como se muestra en la figura 6 son motores de corriente continua sin escobillas, con bajo factor de potencia y un rendimiento inferior al de los motores trifásicos; su rotación se divide en ciertos número de pasos debido a su estructura, por lo que el funcionamiento del motor paso a paso se acompaña de un sonido y vibración característicos (Conti, 2005).

Este tipo de motores recibe órdenes de lazo abierto a través de circuitos digitales que controlan los drivers para aumentar su capacidad de carga a la salida, además no necesitan retroalimentación, envejecen lentamente y tienen buenas características de bloqueo; sin embargo, su control es de mayor complejidad con respecto a los servomotores. Su velocidad es relativamente constante y dependiendo de la frecuencia y de la magnitud de los pasos se puede obtener un movimiento brusco o suave. Los devanados que son la carga en las salidas

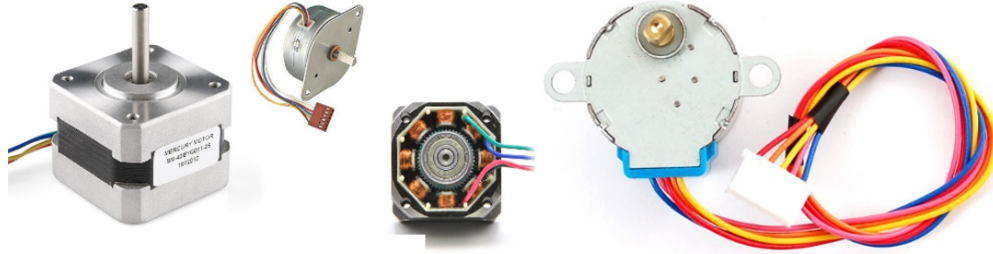


Figura 6: Motores paso a paso

del controlador son bobinas con cierta inductancia y capacitancia. Su reactancia aumenta al aumentar la frecuencia, lo que limita la corriente que fluye y limita la frecuencia de conmutación máxima; además uno de los parámetros principales a considerar al momento de seleccionar un motor paso a paso es el par de retención (Holding Torque) que tiene el motor ya que define la potencia del mismo (Conti, 2005).

4.2.6.1. Torque del motor

El torque o momento de torsión de una fuerza se define como el producto vectorial entre una fuerza \vec{F} y la posición a la que se aplica la fuerza con respecto a un punto de origen o de referencia para el torque \vec{r} ; por lo tanto, la expresión para calcular el torque es el que se muestra en la **Ecuación (9)**; la magnitud del torque tiene la tendencia a producir una rotación ya sea en sentido horario o anti-horario, sin necesidad de enfatizar en el aspecto vectorial del torque (Slisarenko y Rojas, 2021).

$$T = \vec{F} * \vec{r} \quad [Nm] \quad \text{Ecuación (9)}$$

Es decir, si se considera que la fuerza que se ejerce para originar el momento torsor es perpendicular al radio de un disco ($\Theta = 90^\circ$) como se observa en la figura 7, se multiplica el valor escalar de la fuerza por el radio. Además, teniendo en cuenta que la fuerza es la aceleración angular de la masa, se obtiene que el momento torsor es proporcional al producto del momento de inercia con la aceleración angular, como se observa en la **Ecuación (10)**.

$$T = I\alpha = \frac{1}{2}mr^2\alpha \quad [Nm] \quad \text{Ecuación (10)}$$

Donde:

I : Momento de inercia del cuerpo.

m : Masa del cuerpo [kg].

r : Radio del cuerpo [m].

α : Aceleración angular del cuerpo [s^{-2}].

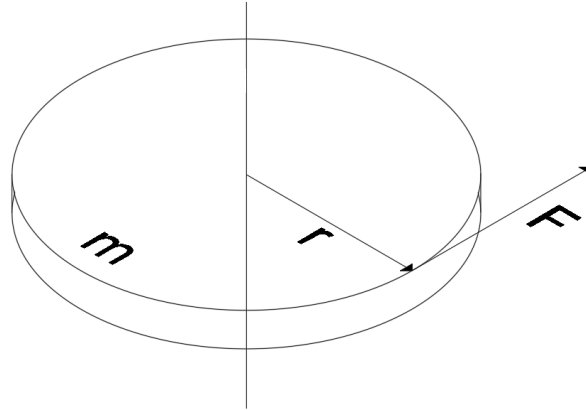


Figura 7: Momento torsor de un disco

4.2.5. Sistema de control lazo abierto

En este sistema el proceso sólo actúa sobre la señal de entrada, sin tener en cuenta lo que ocurra en la señal de salida. Estos sistemas no tienen realimentación, por lo cual la salida no actúa sobre los valores de la señal de entrada, por lo tanto, la salida no se mide ni se realimenta para compararlo con la entrada, es decir la salida no tiene efecto sobre la señal o acción de control. Los elementos que se muestran en la figura 8 integran un sistema de lazo abierto, los cuales se pueden dividir en dos partes: el controlador y el proceso controlado, de manera simple con controlador solo suele ser un amplificador, filtro o unión entre dos mecanismos (Alberto Perez et al., 2017).

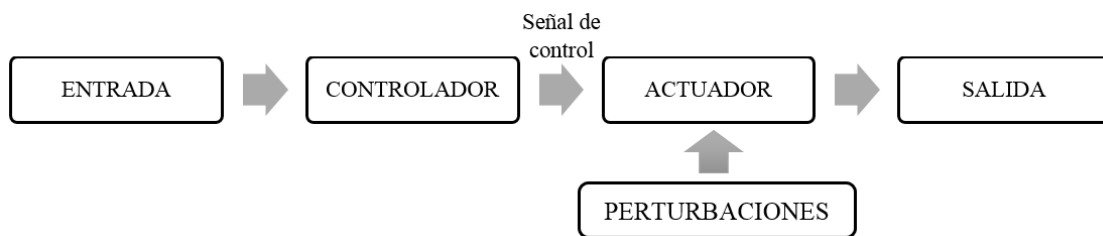


Figura 8: Sistema de control a lazo abierto

Por lo tanto, la exactitud del sistema depende de la calibración, es decir, se establece una relación entre la entrada y la salida con el fin de obtener del sistema la exactitud deseada. Adicionalmente cabe mencionar, que estos sistemas de control son económicos e inestables a perturbaciones, además no puede corregir ningún error y por lo tanto no puede compensar las perturbaciones en el sistema (Alberto Perez et al., 2017).

Como se puede observar en la figura 8, las partes de control de un sistema de lazo abierto son las siguientes (Alberto Perez et al., 2017):

- *Señal de entrada:* Puede ser analógica o digital.

- *Controlador*: Dispositivos encargados de controlar el proceso.
- *Actuador*: Planta o proceso controlado.
- *Perturbaciones*: Señales no deseadas que afectan al funcionamiento del sistema.
- *Salida*: Resultado final del sistema.

4.2.6. Integrated Development Environment (IDE)

El entorno de programación IDE es un software que contiene el entorno de desarrollo; está compuesto por un conjunto de herramientas de programación, cuyo lenguaje de programación puede ser exclusivo o utilizarse en varias aplicaciones. Consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI. Además, incorpora herramientas para cargar el programa ya compilado a la memoria flash del hardware.

Los ficheros están compuestos con la extensión “ino” y deben estar en una carpeta con el mismo nombre del fichero.

4.2.7.1. Entorno de programación

La versión 1.6 del IDE cuenta con soporte multiplataforma de Arduino, detección automática de la placa conectada, muestra memoria Flash y SRAM ocupada por el proyecto, autoguardado al compilar y cargar proyecto. Algunos de los componentes con los que cuenta el entorno de desarrollo son los que se describen en la figura 9 y se destacan los siguientes (Crespo, 2016).

- Botones de Verificar y Subir.
- Botón Monitor Serie.
- Consola de Error.
- Menú herramientas Placa y Puerto.
- Puerto y placa seleccionada.
- Menú preferencias.
- Proyecto/Sketch.
- Resaltado de palabras clave.

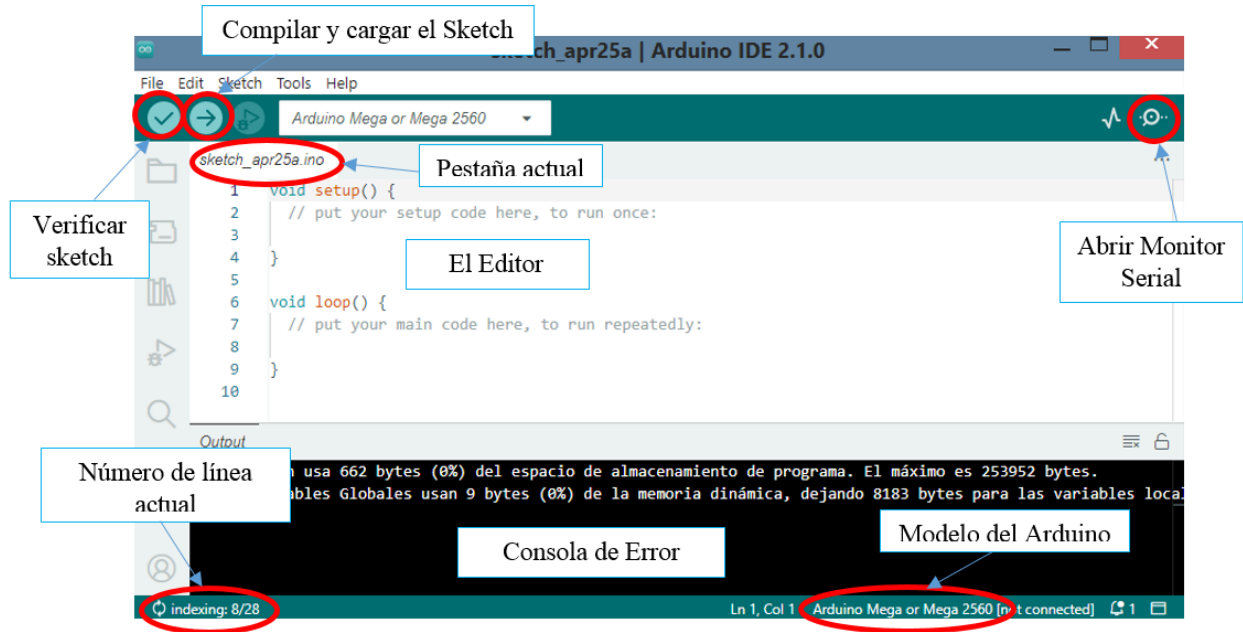


Figura 9: Entorno de trabajo IDE

4.2.7. Módulo ESP32

Es un controlador que funciona a 3.3 V, el cuál al ser adaptado a una tarjeta de desarrollo se lo puede llevar a múltiples aplicaciones, integra funciones como wifi y bluetooth, su frecuencia de trabajo se encuentra en 2.4 GHz de 32 bits, con potencia computacional, I/Os, RTC, soporte para operaciones diversas comunicaciones y bloques dedicados a la seguridad. Según su datasheet, las especificaciones técnicas que tiene la tarjeta son (Babiuch et al., 2019):

- Wifi 802.11 b/g/n.
- Soporte para WMM y funcionamiento simultáneo como modo Station y SoftAP.
- Bluetooth v 4.2 BR/EDR y BLE, transmisión 12 dBm y recepción 97 dBm y con comunicación UART.
- Microprocesador Xtense/dual core 32 bits, con 448 kb de ROM y soporte de memoria flash y SRAM.
- Oscilador interno de 8 mHz y externo de 2 MHz hasta 60 MHz.

4.2.8.1. DEVEKIT V1

Es una placa que dispone de 36 pines para su trabajo, cuenta su propio puerto USB para llevar a cabo su programación o suministrar energía. Su voltaje de alimentación es de

5 V, pero todos sus pines de entradas y salidas funcionan a 3.3 V; por otro lado, cuenta con memoria interna destinadas al procesamiento y arranque del procesador y memoria externa de 4 GB destinadas al expandir la memoria del código, como se muestra en la figura 10.

Los pines de la tarjeta se encuentran descritos en la figura 10 y al igual que en la placa de Arduino nos permite conectar varios dispositivos como relés, sensores, entre otros instrumentos; así mismo tiene convertidores analógicos para la lectura de sensores y generadores de PWM para dar señales analógicas. Además, cuenta con puertos UART o comunicación serial para comunicaciones asíncronas para dos pines TX y RX; además dispone de 2 puertos I2S para transmisión de audio. Por último, cuenta con aceleradores de algoritmos orientados a la encriptación (Bertoleti, 2019).

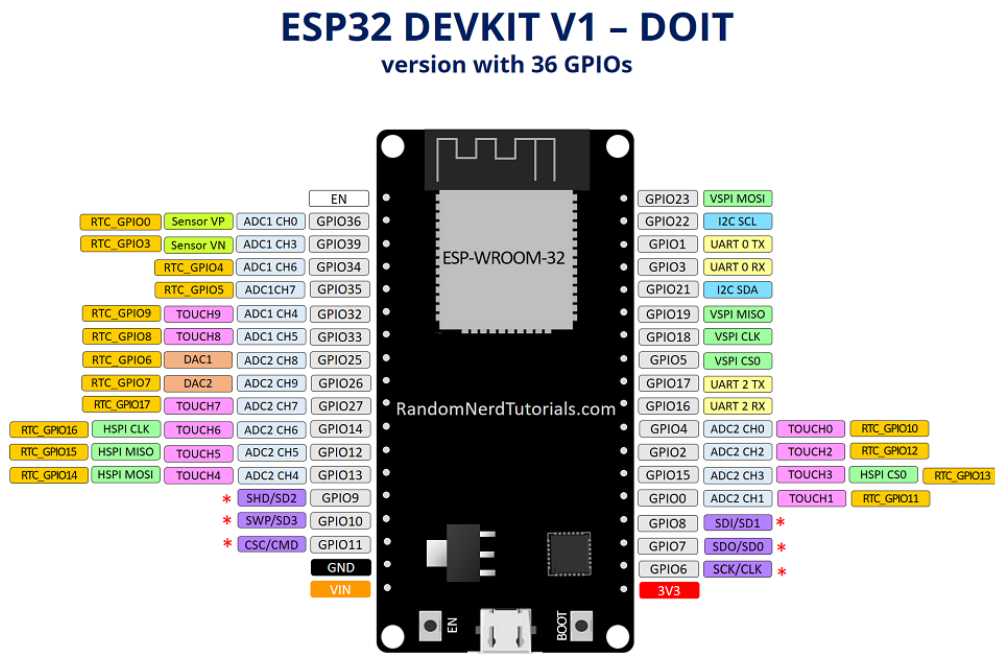


Figura 10: Tarjeta ESP32

4.2.8. Comunicación Inter Integrated Circuit (I2C)

La comunicación I2C es un protocolo desarrollado para comunicar varios chips, cuya tecnología se ha ido desarrollando con el paso del tiempo hasta estandarizarse a nivel mundial. Se utiliza para la comunicación de múltiples sensores y actuadores por medio de solo dos pines, debido que el protocolo permite velocidades de 100, 400 y 1000 Kbits/s; los mensajes que se envían por este puerto incluyen información y una dirección tanto del registro como del sensor. Por lo tanto, la información enviada requiere la confirmación de recepción parte del dispositivo controlador (Sánchez, 2022).

4.3 Capítulo III: Android Studio

Es un entorno de desarrollo integrado (IDE) oficial que se usa en el desarrollo de apps para Android. Basado en el potente editor de código y herramientas para desarrolladores de IntelliJ IDEA. Además, ofrece funciones que mejoran la productividad al momento de compilar apps para Android, como las siguientes (Android, 2023):

- Un sistema de compilación flexible basado en Gradle; el cuál es una herramienta de depuración que trabaja en varios lenguajes de programación, porque Android Studio no cuenta con un gestor para este proceso y Gradle lo permite realizar con un alto rendimiento y bajo una amplia librería de dependencias.
- Un emulador rápido y cargado de funciones.
- Un entorno unificado desarrollado para todos los dispositivos Android.
- Ediciones en vivo para actualizar elementos componibles en emuladores y dispositivos físicos, en tiempo real.
- Integración con GitHub y plantillas de código para compilar funciones de apps comunes y también importar códigos de muestra.
- Variedad de marcos de trabajo y herramientas de prueba.
- Herramientas de Lint para identificar problemas de rendimiento, usabilidad y compatibilidad de versiones, entre otros (Android, 2023).
- Compatibilidad con C++ y NDK.
- Compatibilidad integrada con Google Cloud Platform, que facilita la integración con Google Cloud Messaging y App Engine.

4.3.1. Estructura de proyectos

Cada proyecto de Android Studio incluye uno o más módulos con archivos de código fuente y archivos de recursos. Entre los tipos de módulos, se incluyen los siguientes (Robledo, 2018):

- Módulos de apps para Android.
- Módulos de bibliotecas.
- Módulos de Google App Engine.

De manera predeterminada, muestra los archivos del proyecto en la vista de proyecto de Android, como se ve en la figura 11. La cuál está organizada en módulos para acceder rápidamente a los archivos fuente clave de cada proyecto. Se pueden visualizar todos los archivos de compilación en el nivel superior de las Secuencias de comandos de Gradle (Robledo, 2018). Los módulos de la app contienen las siguientes carpetas:

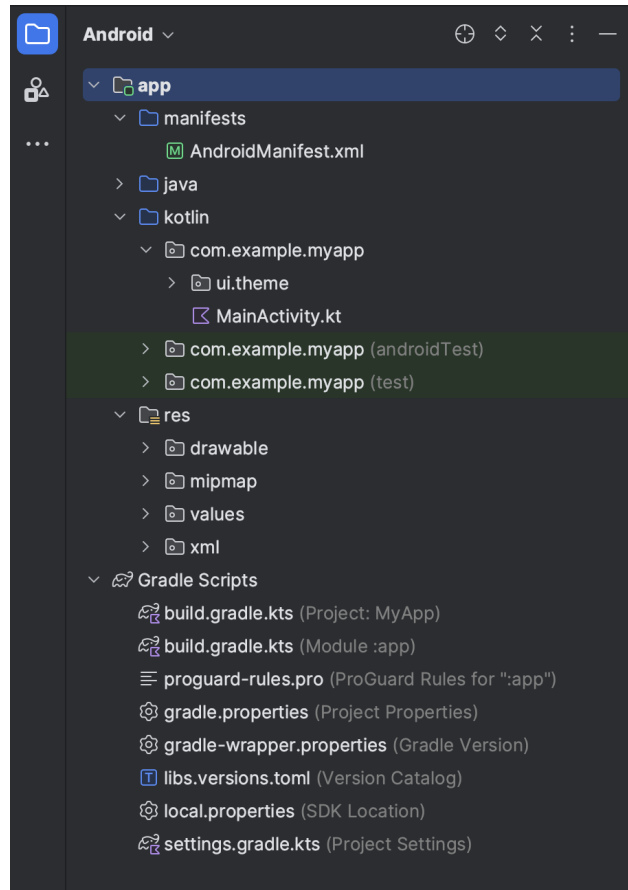


Figura 11: Archivos de proyectos en la vista de proyectos de Android

- *Manifests*: Contiene el archivo AndroidManifest.xml.
- *Java*: Contiene los archivos de código fuente de Kotlin y Java, incluido el código de prueba JUnit.
- *Res*: Contiene todos los recursos sin código, como cadenas de IU y, además, imágenes de mapa de bits.

4.3.2. Herramientas de depuración y perfil

4.3.2.1. Depuración integrada

Mejora las explicaciones de código en la vista del depurador, debido a que verificar de manera integrada los valores de las variables integradas, los objetos que hacen referencia a un objeto seleccionado, los valores de retorno de métodos, las expresiones lambda y de operados, y los valores del cuadro de información (Samsung, 2023).

4.3.2.3. Inspección de código

Al momento de compilar un programa, Android Studio ejecuta de manera automática las inspecciones de lint y otras inspecciones de IDE para identificar y corregir problemas con la calidad estructural de manera sencilla. La herramienta lint comprueba los archivos de origen del proyecto de Android en busca de posibles errores, para realizar mejoras relacionadas con la precisión, seguridad, rendimiento, usabilidad, accesibilidad e internacionalización (Samsung, 2023).

Además de las verificaciones de lint, Android Studio realiza inspecciones de código de IntelliJ y valida anotaciones para simplificar el flujo de trabajo de codificación (Samsung, 2023).

4.3.3. Entorno de desarrollo de Android

Este software es un sistema multiplataformas, cuyo entorno de desarrollo está compuesta por múltiples herramientas como se puede visualizar en la figura 12, en donde se encuentran elementos como (Android, 2023):

1. *Barra de herramientas*: Permite realizar una gran variedad de acciones, como ejecutar la app e iniciar las herramientas de Android.
2. *Barra de navegación*: Proporciona una vista más compacta de la estructura visible en la ventana Project.
3. *Ventana del editor*: Crea y modifica el código, además según el tipo de actividad actual, el editor puede cambiar.
4. *Barra de ventanas de herramientas*: Se puede expandir y contraer mediante los botones del exterior.
5. *Ventanas de herramientas*: Brinda acceso a tareas específicas, como la administración de proyectos, la búsqueda, el control de versiones, entre otras.
6. *Barra de estado*: Muestra el estado del proyecto y el IDE, además de advertencias o mensajes.

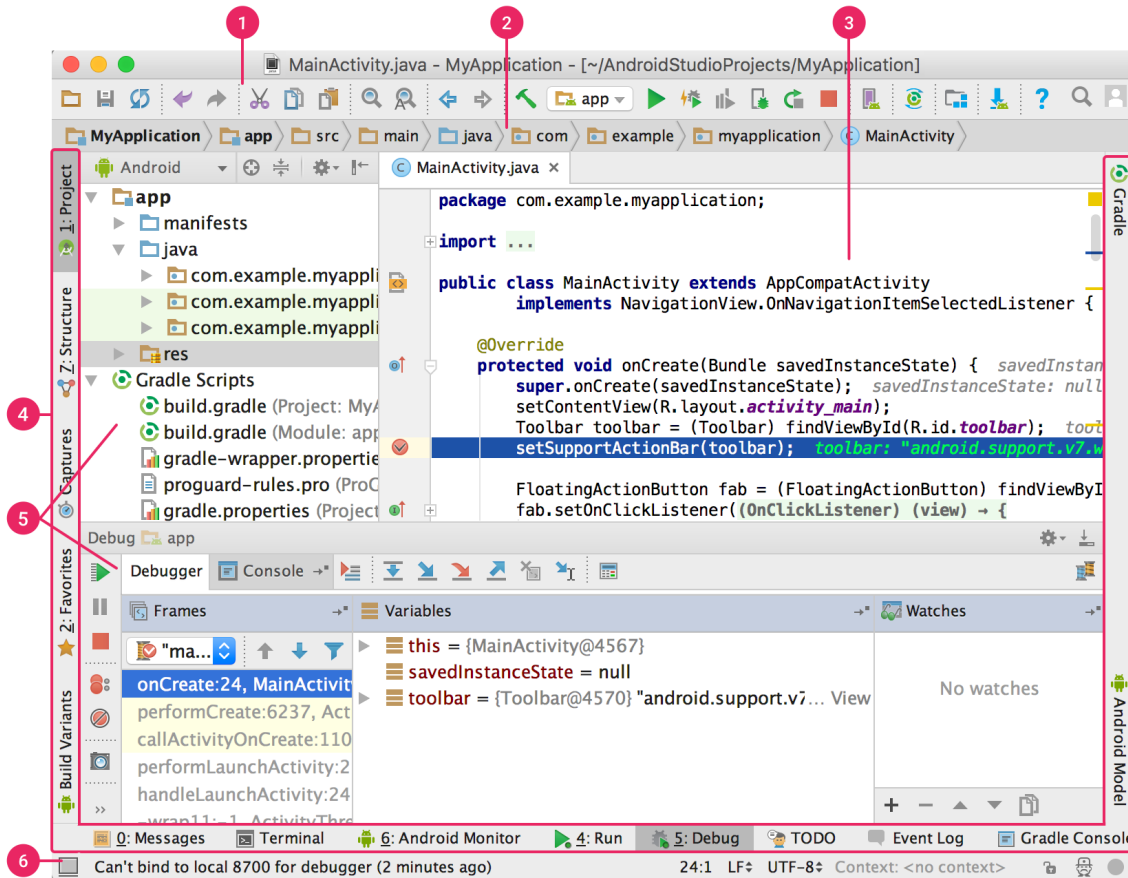


Figura 12: Interfaz Android Studio

4.3.4. Lenguaje de programación Java

Un lenguaje de programación es un conjunto de sintaxis y reglas semánticas que definen los programas de computadora y mediante el cual se le pueden entregar instrucciones. Estas instrucciones son entendibles por la computadora y no necesitan traducción para comprender y ejecutar el programa (Roman Arenaza, 2019).

Java es un lenguaje de programación que se creó a finales del año 90, el cual está totalmente orientado a objetos, debido a los inconvenientes que generaba C++ con estas estructuras. Algunas de sus principales características es que es un lenguaje robusto, independiente de plataformas y multiárea, el cual soporta características propias de paradigmas de la programación como: encapsulación, herencia y polimorfismo (Roman Arenaza, 2019).

De la misma forma, un algoritmo es un conjunto de pasos estructurados, que cuenta con reglas no ambiguas que le permiten dar soluciones a problemas determinados. Por otro lado, en la programación orientada por bloques se organizan como colecciones cooperativas de objetos, los cuales representan una instancia de alguna clase y estas clases están unidas mediante relaciones de herencia (Roman Arenaza, 2019). Además, Java realiza verificaciones

en busca de plataformas tanto en tiempo de compilación como en tiempos de ejecución, el cual mediante la comprobación ayuda a detectar errores durante el ciclo de desarrollo (Roman Arenaza, 2019).

En el caso de Android Studio, Java es un lenguaje popular para el desarrollo de entornos multimedia debido a su potencia, versatilidad y multiplataformas. Es un lenguaje que sirve para el desarrollo de un entorno, sistemas operativos móviles y es utilizado para crear aplicaciones en él. En la tecnología de Java es un programa que se puede incrustar en un documento HTML “página web”, esto nos ayuda a crear programas donde el usuario lo podrá ejecutar, con acceso a bases de datos que están orientadas al mundo empresarial (Martinez Vaca, 2021).

4.3.5. JavaScript Object Notation (JSON)

JSON es un formato ligero basado en los tipos de datos del lenguaje de programación JavaScript. Estos diccionarios consisten claves y valores, cuyo valor permite un nivel arbitrario de anidamiento, además de los diccionarios simples, JSON admite matrices y tipos atómicos, como números y cadenas, con formatos compositivos que pueden contener a otros documentos JSON (Bourhis et al., 2020).

Es simple y fácilmente legible, por lo que se convierte en uno de los formatos más populares para intercambiar datos en la Web. Como es en el caso de los servicios web, que se comunican con sus usuarios a través de una interfaz de programación de aplicaciones (API), esto debido a que JSON es un formato predominante para enviar solicitudes y respuestas de API a través del protocolo HTTP. Además, el formato JSON se usa mucho en sistemas de bases de datos (Bourhis et al., 2020).

Actualmente la literatura de investigación es muy escasa y todavía no hay acuerdo sobre el marco teórico correcto para JSON. Sin embargo Bourhis et al., 2020, define el modelo de datos, examinando las características clave de los documentos JSON y cómo se utilizan en la práctica, dando como resultado estructura en forma de árbol muy similar al modelo de árbol de datos ordenados de XML, pero con algunas diferencias clave, por ejemplo los árboles son deterministas por diseño debido al nivel de complejidad del lenguaje JSON, las matrices están explícitamente presentes en JSON, lo que no ocurre en XML, la característica definitoria de cada diccionario JSON es que no está ordenado y finalmente, los valores JSON son nuevamente objetos JSON (Bourhis et al., 2020).

4.3.6. Firebase

Firebase es una plataforma de desarrollo de apps que ayuda a compilar y desarrollar aplicaciones con el respaldo de Google, tanto para Android, iOS y desarrollo web. Para ello, genera herramientas para un desarrollo más rápido y de calidad. Dentro de los servicios que provee están (Diaz, 2021):

- *Firestore Analytics*: Es un servicio que permite a los desarrolladores entender cómo los usuarios están usando la aplicación a partir de un SDK que captura eventos y propiedades por sí solo y a la vez permite obtener datos customizados.
- *Firestore Cloud Messaging*: Es un servicio que permite enviar mensajes y notificaciones para Android, Aplicaciones web e IOS.
- *Firestore Auth*: Es un servicio que permite autenticar usuarios usando solo código cliente. Soporta proveedores de login como Facebook, Google, Gitgub y Twitter. También incluye un sistema gestor de usuarios.
- *Firestore Real-time Database*: Es un servicio que permite tener una Base de datos a tiempo real sincronizada a través de todos los usuarios y almacenada en la nube. Permite integración con Android, IOS y aplicaciones JavaScript.
- *Firestore Storage*: Es un servicio que incrementa la seguridad y facilita la transferencia de ficheros sin importar la calidad de la red.
- *Firestore Test Lab for Android*: Es un servicio que provee de una infraestructura basada en la nube para el testeo de aplicaciones Android.
- *Firestore Crash Reporting*: Es un servicio que permite consultar los reportes de errores creados en la aplicación de forma detallada.
- *Firestore Notifications*: Es un servicio que permite enviar notificaciones para cierto objetivo de usuarios (Diaz, 2021).

4.3.7. Realtime Database

La plataforma de Firebase, suministra una base de datos organizada de back-end en tiempo real. La cuál facilita a los desarrolladores de aplicaciones una API, que sincroniza la información de la aplicación y almacena los datos en la nube. Para ello se requiere de un usuario y contraseña para establecer una conexión estable, además usa un caché local

en el dispositivo, por lo que no se pierde información de ningún tipo. En el caso de pérdida de conexión los datos se sincronizan automáticamente con los datos locales sincronizan automáticamente con sus datos locales Villalón Pardo, 2021.

Los datos se almacenan en formato JSON y se sincronizan en tiempo real con cada cliente conectado. Todos los clientes comparten una instancia de Realtime Database y reciben actualizaciones automáticamente con los datos más recientes. Firebase Realtime Database tiene una forma de almacenar datos NoSQL, por lo que no utiliza tablas para guardar los datos, sino que utiliza JSON por lo que no utiliza Query para insertar, actualizar, eliminar o añadir información. Es considerada la parte back-end de un sistema para almacenar datos en una base de datos Villalón Pardo, 2021.

Cabe mencionar que la plataforma proporciona reglas de seguridad para definir la estructura de los datos y cómo se van a leer o escribir. Para ello hace uso de un acceso mediante el código del cliente, por lo que persisten de forma local si se pierde o no la conexión.

5. Metodología

5.1 Área de estudio

La zona de aplazamiento del presente proyecto de investigación será en la Universidad Nacional de Loja; en el bloque 3 de la Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables, el cual está ubicada en el Sur de la ciudad de Loja, en las Av. Reinaldo Espinoza y Eduardo Kigman, tal y como se observa en la figura 13, con las coordenadas geográficas $4^{\circ}01'49''S$ $79^{\circ}11'58''W$.

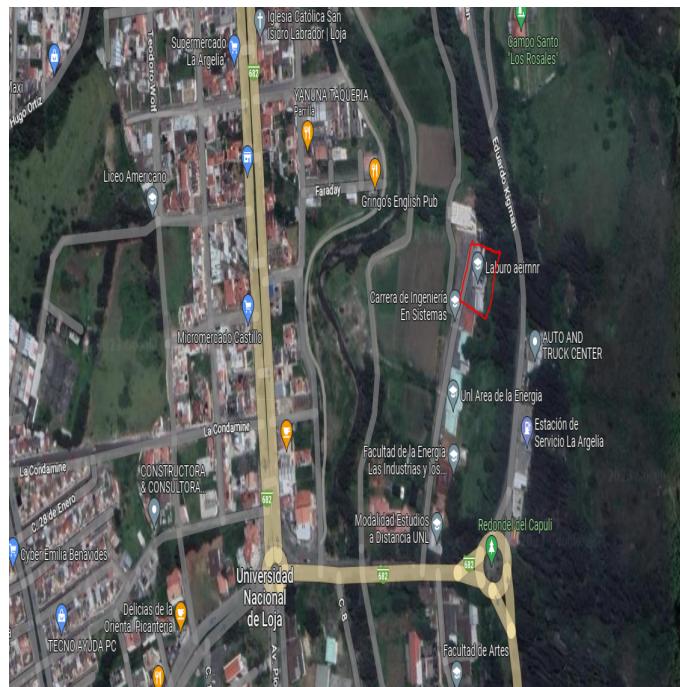


Figura 13: Ubicación geográfica del lugar de estudio

Fuente: (Maps, 2023)

5.2 Equipos y materiales

5.2.1. Equipos

- Computadora.
- IDE Arduino.
- Paquete de Microsoft Office.

- Arduino Cloud.
- Android Studio.

5.2.2. Materiales

- Catálogos.
- Manuales de luminotécnica y libros de iluminación.
- Tesis y Artículos científicos.
- Libros de automatización en arduino.
- Los materiales requeridos para la construcción del prototipo que se muestran en la tabla 3.

Tabla 3: Materiales requeridos para el prototipo

N°	Descripción	Cantidad
1	Módulo ESP-32 WROOM/Dev. Module	1
2	BH1750	2
3	Cable UTP Cat.06	2 m
4	Servomotor 180°	1
5	Motor Paso a Paso	1
6	Placa PCB Perforada para soldar	1
7	Rollo de estaño	1
8	Módulo J298N	1
9	Persianas 1,5 m x 4 m	1
10	Botones iluminados	6

5.3 Procedimiento

5.3.1. Primer objetivo: Identificar los niveles de iluminación requeridos para el edificio 3 de la Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables.

Para ello se aplicó el procedimiento descrito de la figura 14

- Haciendo uso de las fuentes bibliográficas disponibles; se recopila la información necesaria para la base argumentativa de la investigación.

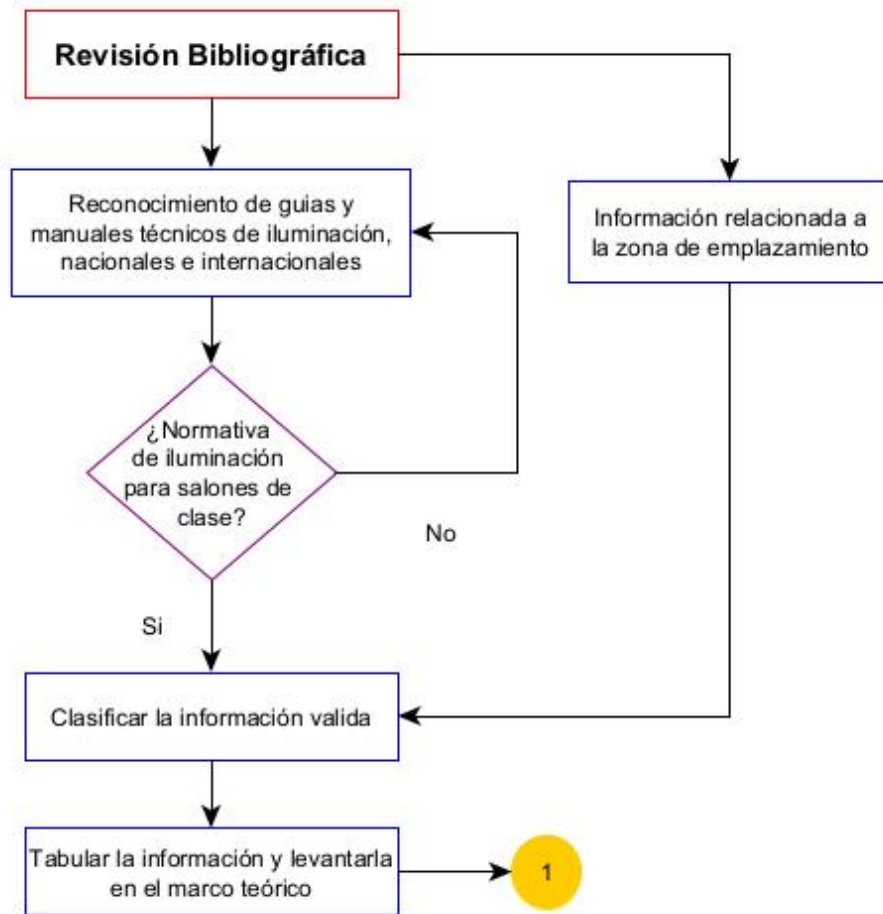


Figura 14: Flujograma de metodología del primer objetivo

- Tomar un conjunto de información técnica relacionadas con niveles de iluminación para un aula de clases, sistemas IoT y la tarjeta Esp32; con la finalidad de conocer y describir las variables, procesos y acciones que se necesitan para automatizar una persiana.
- Redactar la información correspondiente en el apartado del marco teórico, con la finalidad de argumentar sólidamente las bases de la investigación.
- Con la información técnica recopilada de los diferentes manuales y normativas, se establece los niveles mínimos de iluminación que se necesitan para espacio de aprendizaje o aula. Además, se selecciona el nivel de iluminación que más se repite para establecerlo como valor estándar y el nivel de iluminación más bajo de las normativas para establecerlo como valor mínimo. Esta escala será importante más adelante para efectuar la programación.
- Con la ayuda del luxómetro BH1750 y una placa Arduino, se realizan las mediciones

de los niveles de iluminación en la zona de emplazamiento, en la parte exterior de la iluminación natural directa y de la parte interior que recibe el aula. Para ello se tomará en cuenta el mallado que establece la norma UNE-EN 12464-1 y la hoja de proceso para el levantamiento de la información.

5.3.2. Segundo objetivo: Diseñar un prototipo de persianas automáticas que regulen el nivel de iluminación en las aulas.

Para el cumplimiento del segundo objetivo se siguió la metodología planteada en el flujograma de la figura 15.

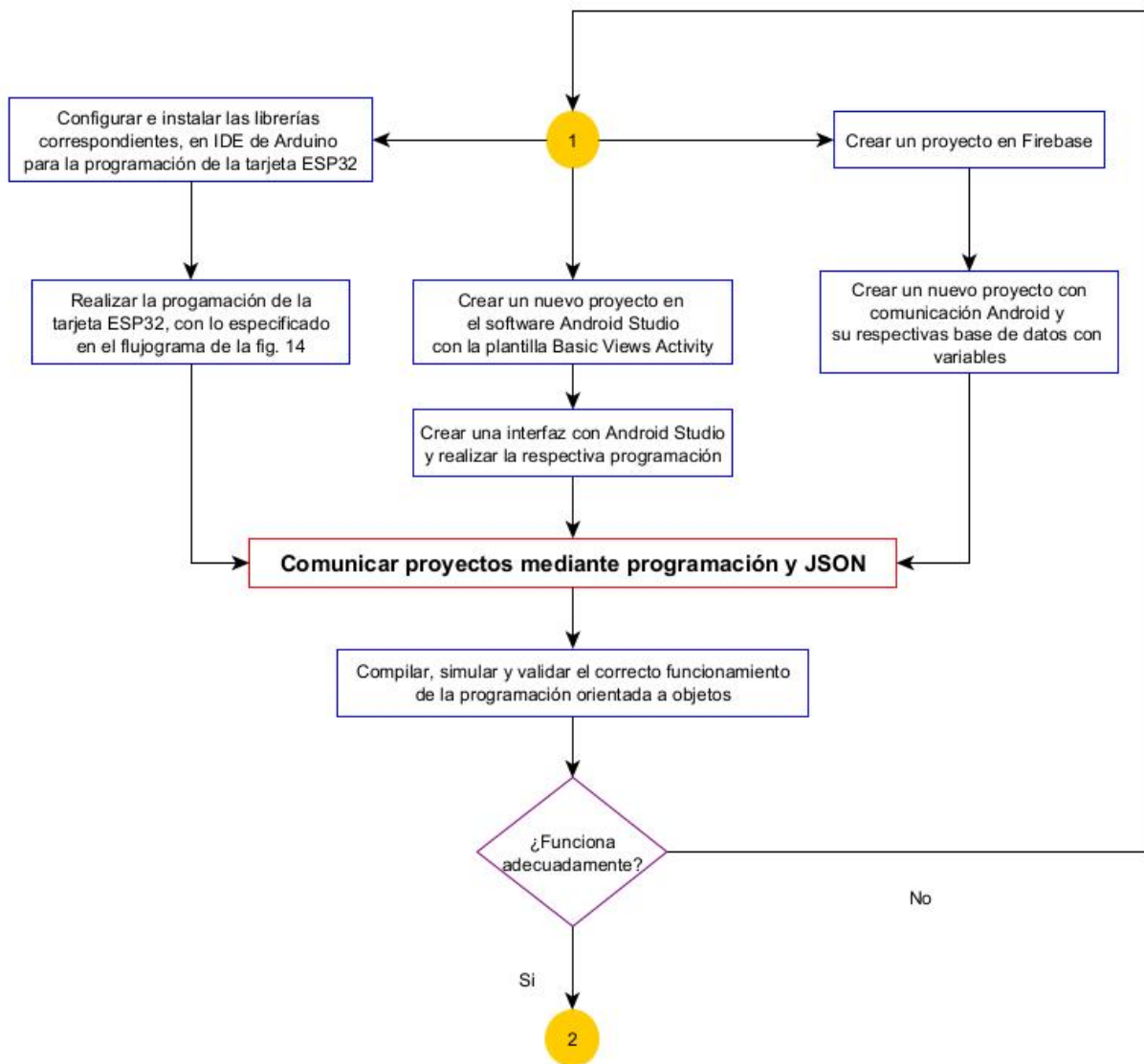


Figura 15: Flujograma de metodología del segundo objetivo

- Para iniciar con la programación del sistema de forma manual y automática; se tomará en cuenta el flujograma de programación mostrado en la figura 16 y el entorno de programación de arduino IDE. Para poder efectuar la programación se debe seguir el siguiente procedimiento:

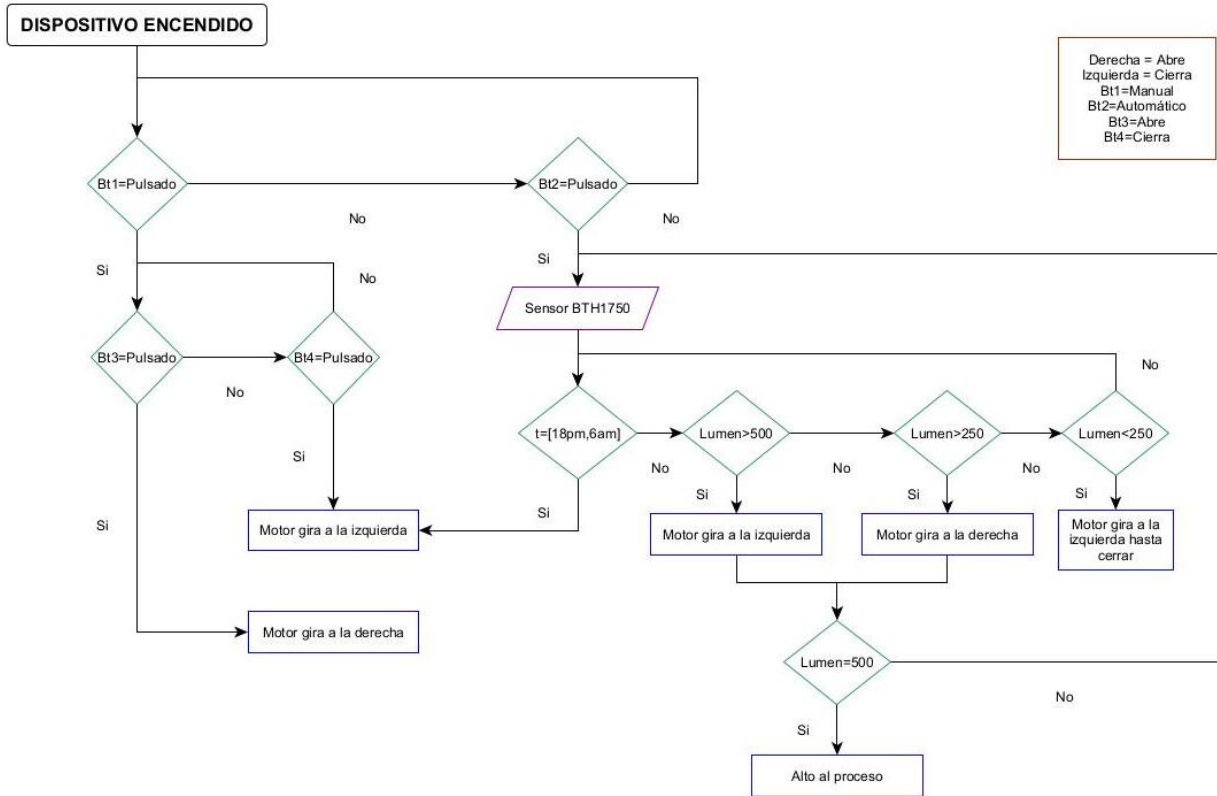


Figura 16: Flujograma de programación del prototipo

1. Conectar la tarjeta a la computadora mediante el cable USB.
2. Abrir el IDE de Arduino versión 2.1.0 y en apartado de File del menú principal seleccionar la opción de preferencias.
3. En la parte de “Additional boards manager URL” pegar la siguiente dirección: <https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package-esp32-index.json> y darle en el botón ok.
4. Seguidamente, en el listado de opciones de la parte izquierda abrimos el administrador de placas y buscamos ESP32, hasta que nos aparezca la opción de la figura 17 y seleccionamos la opción de instalar.

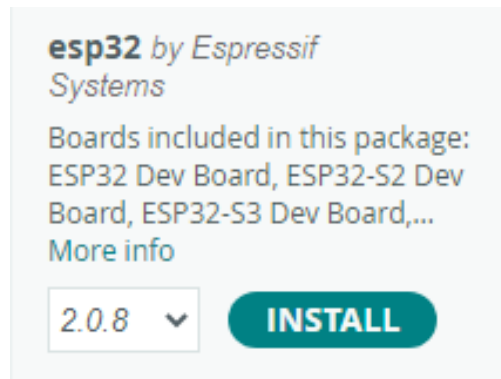


Figura 17: Librería a implementar para la tarjeta

5. Una vez instalado, en el caso de ser necesario se procede a seleccionar la placa con la cual se va a trabajar, en este caso es la “ESP32 Dev Module”; de la misma forma se puede cambiar las configuraciones de las placas según los requerimientos.
6. Para cargar el proyecto a la tarjeta se debe seleccionar la opción de cargar y se debe mantener presionado el botón de BOOT de la placa hasta que aparezca el mensaje de conexión el IDE de arduino, soltamos el botón y esperamos a que se cargue el código.
 - Para iniciar con la programación de la aplicación en el software de Android Studio, en primer lugar, se debe crear un proyecto considerando utilizar la plantilla de "Basic Views Activity", de la misma forma se debe asignar el nombre del proyecto, escoger el lenguaje de programación "Java", y el sistema operativo "(API 22: Android 5.1 (Lollipop))", debido a que es uno de los sistemas más comunes en los dispositivos móviles.
 - Establecer la configuración con Firebase; por lo cual, al ser un servicio dado por la compañía GOOGLE, se debe ingresar al servicio con una cuenta de la compañía. Seguidamente se debe redireccionar el usuario a la pestaña de ir a consola, crear un proyecto y seleccionar la opción de Android. En el proyecto se debe ingresar la información de la aplicación como el package y el certificado de firma SHA-1.
 - Una vez creado el proyecto en Faire Base, se procede a realizar la vinculación con la aplicación en desarrollo; para ello desde la aplicación en desarrollo de Android Studio, se procede a ingresar los datos de la Firebase y pegar el archivo JSON correspondiente en el proyecto de la Android Studio. Para ello se debe implementar las dependencias del proyecto en Firebase, así como los plugins en los gradle tanto para app y Project, respectivamente. Se finaliza el proceso sincronizando nuevamente los gradles.
 - Al contar Firebase con múltiples subservicios, se procede a seleccionar la opción de Realtime Database y definir las reglas en valor "True". A continuación, se procede a

ingresar las variables con sus respectivos valores iniciales tanto para los botones de funcionamiento como para los valores leídos por el sensor.

- Se agrega a la programación de IDE Arduino, lo correspondiente a la comunicación con la Firebase y la lectura y escritura de los valores de las variables de la base de datos.
- Con el archivo "Principal Activo (Active Main)" de la carpeta "layout" se procede a realizar el diseño de la interfaz de la aplicación, considerando el número de botones y el indicador del nivel de iluminación.
- En el archivo "Principal Activo (Active Main)" de la carpeta "Java" se procede a realizar la programación de todos los objetos de la aplicación, para ello se debe importar las librerías de cada objeto con sus respectivas etiquetas y realizar la programación de la acción de cada botón y el indicador.
- Finalmente, se realiza la compilación e instalación de la aplicación en el dispositivo móvil, considerando que se debe conectar el teléfono celular en modo programador vía USB y con la opción habilitada de "Depuración vías USB".

5.3.3. Tercer objetivo: Construir el prototipo diseñado para validar su correcto funcionamiento.

Para el cumplimiento del segundo objetivo se siguió la metodología planteada en el flujograma de la figura 18.

- Se arma el prototipo en un protoboard con la finalidad de verificar la funcionalidad de la programación y comprobar la funcionalidad de cada objeto; para ello se debe tener en cuenta que el sensor de iluminación BH1750 debe conectarse a los puertos de comunicación I2C, es decir la entrada SCL del pin 22 y SDA del pin 21.
- Después de haber establecido las variables, se procede a identificar los componentes que ayuden automatizar un prototipo de persiana; para ello se tomará en cuenta los sensores, actuadores, botones, tarjeta de control y el sistema de alimentación, que influirán en el mismo, así mismo se realiza las listas de materiales con sus respectivos costos para obtener el costo final del prototipo.
- Tomando en cuenta los requerimientos para el prototipo deseado, se selecciona los sensores y actuadores necesarios para monitorear y controlar el nivel de iluminación de un aula del bloque tres de la facultad de la Energía, las industrias y los Recursos Naturales no Renovables. Seguidamente con la ayuda de una placa PCB perforada y cable UTP,

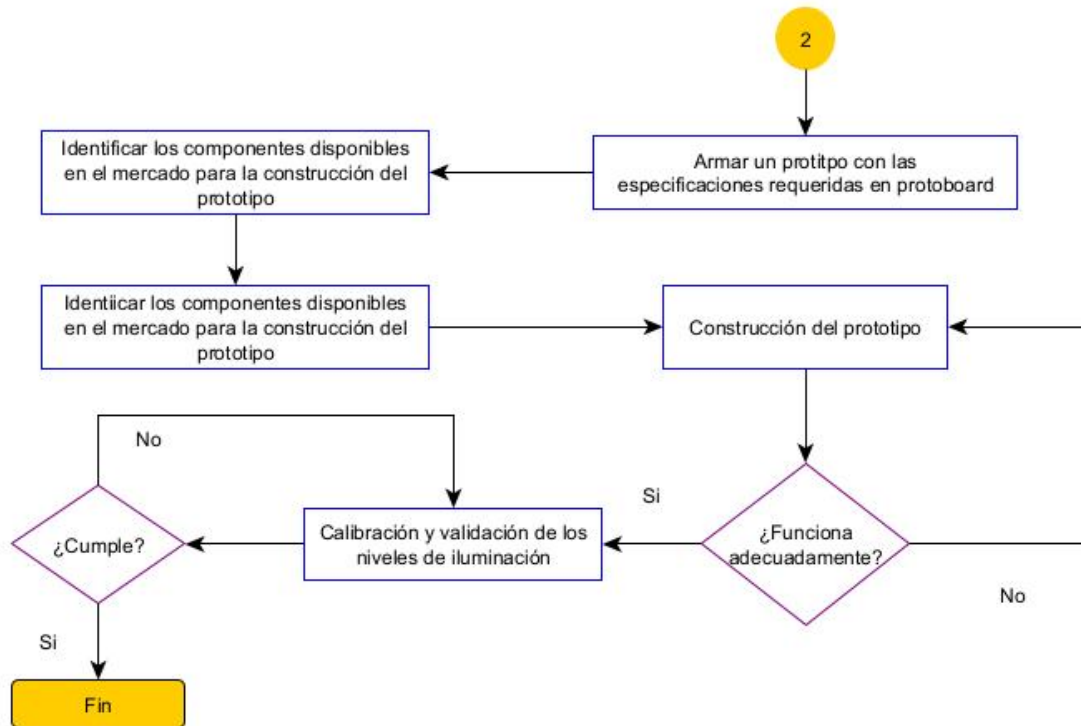


Figura 18: Flujograma de programación del prototipo

se procede a realizar la construcción, armado e implementación del dispositivo para las primeras pruebas de funcionamiento.

- Se realiza la validación del dispositivo, midiendo el nivel de iluminación al interior del aula y procurando el nivel de iluminación corresponda a lo establecido en la normativa

5.4 Procesamiento

5.4.1. Primer objetivo: Identificar los niveles de iluminación requeridos para el edificio 3 de la Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables.

- Con el equipo de medición y la práctica experimental diseñada y ejecuta; se procedió a tabular toda la información recopilada.
- Las mediciones al ser realizadas por cada malla durante 5s, se calculó el promedio de nivel de iluminación por cada malla.
- Se tabulo la información anteriormente obtenida y con la ayuda del Excel se procedió a originar la gráfica de calor correspondiente.

5.4.2. Segundo objetivo: Diseñar un prototipo de persianas automáticas que regulen el nivel de iluminación en las aulas.

- Considerando las especificaciones técnicas que requiere el prototipo, se identificó los materiales disponibles para su construcción.
- Se seleccionaron los materiales necesarios, considerando que el mecanismo de control debe ser lo más compacto posible.
- Se tabularon todos los elementos seleccionados con su respectivo costo para identificar el costo de inversión del prototipo

5.4.3. Tercer objetivo: Construir el prototipo diseñado para validar su correcto funcionamiento.

- Una vez que el prototipo ha sido implementado en la zona de emplazamiento, se procedió a realizar las mediciones con respecto a los niveles de iluminación exterior, iluminación interior después de la persiana y niveles de iluminación en un punto medio de la superficie del aula.
- Se tabuló la información recopilada y se realizó la gráfica de Iluminación Exterior vs Iluminación Interior ya sea después de la persiana y en un punto medio de la superficie del aula.
- Con la ayuda del software Excel se obtuvo las funciones lineales que describen la gráfica.

6. Resultados

6.1 Levantamiento de la información

6.1.1. Mallado de la zona de emplazamiento

Considerando que las distancias del aula de emplazamiento son de 6.13 m por 7.58 m, se procedió a realizar el cálculo de las dimensiones de las mallas mediante la **Ecuación (3)**, obteniendo el siguiente resultado:

$$p = 0.2 * 5^{\log_{10}(7.58m)} = 0.82 \text{ m}$$

Por lo tanto, el distributivo de las mallas es el que se muestra en la figura 19, conllevando a realizar 72 mediciones de iluminación en el aula A-3-2-2, en la mitad del día con respecto a la hora solar efectiva; para ello aplicando la **Ecuación (5)** para todos los días de la semana se obtuvo los resultados mostrados en el anexo 2, y el cálculo para la media de todos los días de la semana es el siguiente:

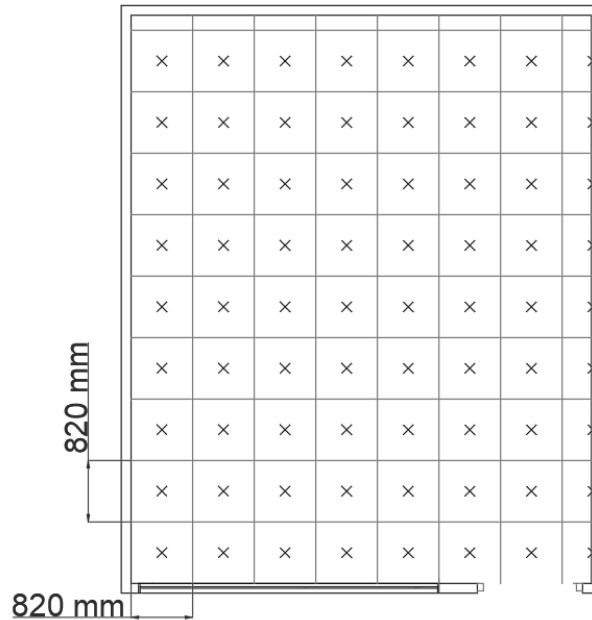


Figura 19: Mallado del área de estudio

$$\delta = \varepsilon_{ob} * \text{sen} \left[\frac{360(284 + n_j)}{365} \right] = 23.45 * \text{sen} \left[\frac{360(284 + 182.5)}{365} \right] = 23.0859$$

El intervalo de tiempo de la hora solar efectiva media según la **Ecuación (6)**, con una latitud de la zona de emplazamiento de -4° y la declinación solar media de 23.0589, es la siguiente:

$$Wt = \cos^{-1}(-\tan(\text{latitud}) * \tan(\delta)) * \left(\frac{24}{360}\right) = \cos^{-1}(-\tan(-4) * \tan(23.0589)) * \left(\frac{24}{360}\right) = 5.886$$

Sin embargo, al realizar el cálculo por cada día del año y deduciendo la media de esos resultados, se obtuvo un valor de 6 horas como se muestra en el anexo, por lo tanto, en promedio de intervalo de tiempo de la hora solar efectiva es de 6 AM hasta 18 PM, como se muestra a continuación según la **Ecuación (7)** y la **Ecuación (8)**:

$$\text{Hora de Salida} = 12 - Wt = 12 - 6 = 6 \text{ h}$$

$$\text{Hora de Puesta} = 12 + Wt = 12 + 6 = 18 \text{ h}$$

6.1.2. Niveles de iluminación

Considerando el distributivo de las mallas de la figura 19 y el periodo de tiempo obtenido al calcular la hora solar efectiva, se procedió a medir los niveles de iluminación del interior y exterior de aula A-3-2-2, tal y como se muestra en el anexo 1, obteniendo la gráfica de calor que se muestra en la figura 20, de la cual se identifica niveles de iluminación superiores a lo establecido por la normativa, pero no inferiores a lo solicitado por las mismas.

282	271,6	285	286,8	278,6	274,8	285	284,8
686,2	603,2	535,2	407	320,8	282	271,6	286,4
825,6	845	600	715,4	678,2	602,4	682,8	691,6
1317,4	1346	1235,2	966,6	877,4	912,2	887	880
1183,4	1220,2	1266,6	1282	1320	1330,6	1242,2	1321,2
1807,2	1775,6	1187,6	1225,2	1927,6	1899,8	1482,8	1338,8
2419,4	2687,2	1911,4	2500,6	2588,4	2419,4	2443	2485,4
4133,4	3046,8	3656,6	3298,4	4133,4	3046,8	3656,6	3298,4
1265,4	3592	2283,2	2759,2	2725,8	3592	2283,2	1298,8

Figura 20: Niveles de iluminación del Aula A-3-2-2

Tomando en cuenta que se tomaron 5 mediciones para cada malla cada 5 segundos, se obtuvo la gráfica de calor de la figura 20, en donde se puede identificar que los niveles de iluminación son superiores a los recomendados por la normativa; por lo cual, calculando el Factor de Luz Diurna (FLD) mediante la **Ecuación (4)** se obtuvo los resultados mostrados en la tabla, del cual la media de este factor es del 3.56 %, por lo tanto, el nivel de iluminación es adecuado al estar en el rango de 3 % a 10 %.

Tabla 4: Nivel de iluminación en la zona de emplazamiento

Número de Fila	Nivel de iluminación interior (lx)	Nivel de iluminación exterior (lx)	FLD
1	2474,95	41510,85	5,94 %
2	3533,80	42926,05	8,23 %
3	2431,85	43607,63	5,58 %
4	1580,58	43680,58	3,63 %
5	1270,78	43796,43	2,90 %
6	1052,73	44359,03	2,38 %
7	705,13	43964,40	1,61 %
8	424,05	43918,53	0,96 %
9	281,08	44078,83	0,64 %
Promedio:	1531,21	43540,6464	3,56 %

6.2. Diseño del prototipo

De acuerdo a los planos de la zona de emplazamiento que se muestran en el anexo 2, se procedió a adquirir una persiana que cuente con las medidas correspondientes a la ventana de la figura 21.

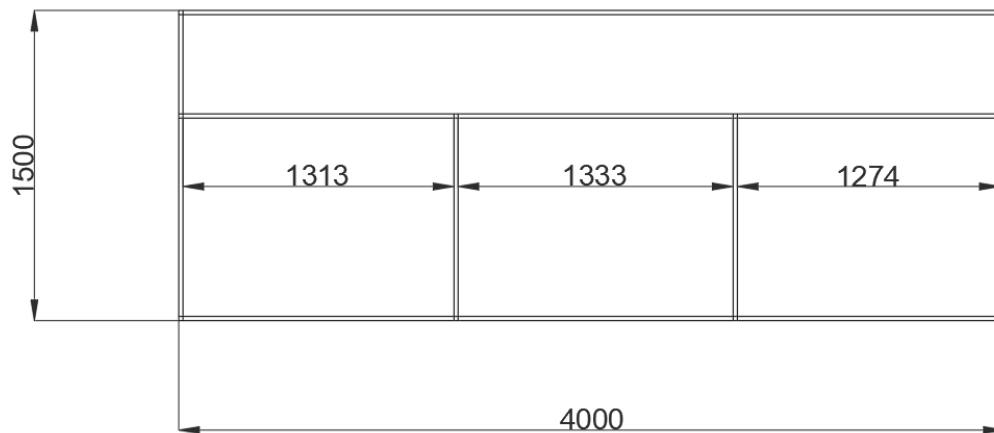


Figura 21: Medidas de la ventana del aula A-3-2-2

Seguidamente se procedió a realizar el diseño del aplicativo móvil para el control de la persiana, para ello se creó un nuevo proyecto en el Software de Android Studio al cual se lo denominó PerIoT y la plantilla base es la Basic Views Activity con un lenguaje de programación Java, además el nombre del paquete (package) del aplicativo es *com.example.periot* y la huella del proyecto es *SHA1: A0:BE:9F:A7:7F:8C:07:33:4D:A3:68:95:92:02:3C:2F:90:FE:A6:92*.

6.2.1. Creación de un proyecto en Firebase

Continuando con la metodología planteada, se procedió a crear un nuevo proyecto en la plataforma de Firebase de Google; para ello se utilizó una cuenta de esta empresa y los datos del package y la huella del proyecto. Se obtuvo el correspondiente archivo JSON de comunicación y se insertaron las siguientes terminales en el Gradle de la aplicación, para su correcta compilación:

- *implementation platform('com.google.firebase:firebase-bom:32.1.1')*: Nos permitió trabajar con una comunicación en Firebase.
- *implementation 'com.google.firebase:firebase-analytics'*: Procesa la información obtenida de Firebase.
- *implementation 'com.airbnb.android:lottie:5.0.1'*: Ejecuta archivos animados de lottie con formato JSON.
- *implementation 'com.google.firebase:firebase-database:20.2.2'*: Conecta con la base de datos del proyecto creado en la plataforma de Firebase

Seguidamente, en la plataforma de Firebase se creó una nueva aplicación Android en el cual se insertó la comunicación con el proyecto y se creó una base de datos en tiempo real (Realtime Database) con las siguientes reglas: todo lo escrito es verdadero (".write":true) y todo lo leído es verdadero (.read":true). Además, en la base de datos se crearon cinco variables con las características mostradas en la tabla 5, tal y como se muestra en la figura 22.

6.2.2. Programación de la tarjeta ESP32

A continuación, se procedió a realizar la programación en la tarjeta ESP32, considerando haber instalado previamente las librerías de reconocimiento de las tarjetas ESP, la librería para trabajar con el sensor BH1750 y la librería de Cliente de Firebase.

El algoritmo inicia insertando las librerías anteriormente descritas, en donde adicionalmente se incluyó la `<Wifi.h>` que permite la conexión a internet por parte de la tarjeta. Seguidamente se define los datos para conectarse a la red y los datos para conectarse a la

Tabla 5: Variables especificadas en la base de datos

Nombre	Tipo de variable	Valor
BOTONABRIR	Boleana	false
BOTONCERRAR	Boleana	false
BOTONMANUAL	Boleana	false
BOTONAUTO	Boleana	false
LECTURAS	Entero	300

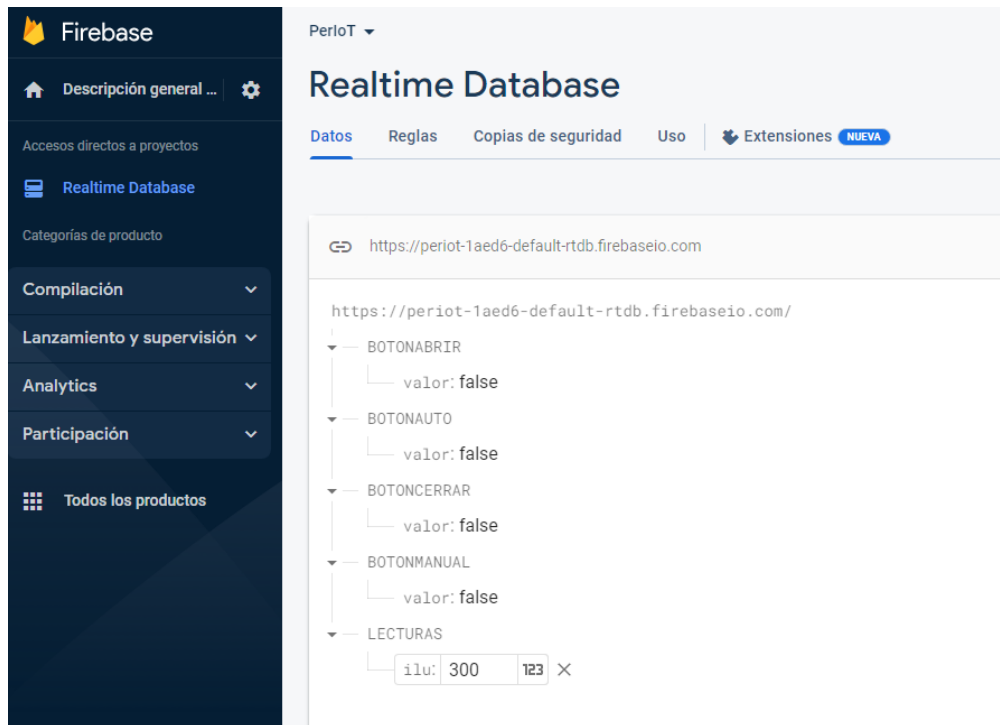


Figura 22: Base de Datos creada en Firebase

base de datos; como la URL que se encuentra en la Base de Datos del proyecto de la Firebase y el secreto generado en las configuraciones del proyecto, en el apartado de secretos de la base de datos.

Luego se establecen las variables que manejará la tarjeta después de leer la información que se lee de la base de datos. Para ello, se consideran cuatro señales de entrada y dos de salida, que nos servirán como indicadores del estado de operación del prototipo. Estas variables son:

- Bt1: Está conectado al pin 12 e indica que el prototipo trabaja de forma Manual.
- Bt2: Está conectado al pin 14 e indica que el prototipo trabaja de forma Automática.
- Bt3: Está conectado al pin 27 e indica que de manera manual la persiana se está

abriendo.

- Bt4: Está conectado al pin 26 e indica que de manera manual la persiana se está cerrando.
- Bt5: Está conectado al pin 32 e indica que la persiana se controla por medio de la aplicación.
- Bt6: Está conectado al pin 34 e indica que la persiana se controla de manera física.
- z: Es una constante booleana en falso.
- y: Es una constante booleana en falso.
- u: Es una constante booleana en falso.
- v: Es una constante booleana en falso. lux: Es una variable flotante en donde se asignan las mediciones realizadas por el luxómetro.

La programación sigue la estructura detallada en la metodología, dando como resultado el algoritmo mostrado en el anexo 3; para ello el programa establece la conexión a la red de internet y a la base de datos, en donde se inserta una reconexión constante con la misma mediante el algoritmo *Firestore.reconnectWifi(true)*.

Cabe mencionar que, el algoritmo de programación trabaja en dos estados, manual y automático; en donde al ejecutarse las acciones de forma manual se habilitan las funciones de los botones abrir y cerrar, y a su vez estos ejecutan acciones en el giro del motor para la persiana. En el caso de ejecutar las acciones de forma automática, el giro del motor está limitado por el nivel de iluminación que indique el sensor; tomando en cuenta los niveles de iluminación de la tabla 1, la normativa DIN 5035 propone el nivel más bajo de iluminación 250 lux, por lo cual se establecerá este valor como el nivel mínimo de iluminación para un aula de una institución académica y según las otras normativas, el nivel de iluminación máximo es de 500 lux; tal y como lo indica la UNE 12464, la NEC y la INEN 1 152.

En el caso que el nivel de iluminación sea más bajo que el límite establecido la persiana se cerrará automáticamente, en el caso que el nivel de iluminación sea mayor a lo recomendado por las normativas, la persiana se cerrará automáticamente de acuerdo a la calibración realizada. En el caso que los niveles de iluminación sean superiores al mínimo establecido (250 lux) pero inferiores al máximo establecido (500 lux); la persiana se abrirá completamente. Finalmente, todo el algoritmo se compiló y cargó a la tarjeta mediante cable USB.

6.2.3. Diseño de la aplicación en Android Studio

Existen muchas formas de integrar un sistema de monitoreo y control de sensores mediante una tarjeta ESP32 y un aplicativo móvil, sin embargo, la mayoría de ellos se encuentran bajo modalidad de pago y las versiones gratuitas cuentan con opciones limitadas; por lo tanto, Android Studio permite realizar entornos IoT con varias herramientas, como el uso de una base de datos externa en Firebase. Para ello en el proyecto anteriormente mencionado (PerIoT), en el apartado de *activity_main* se procede a diseñar la interfaz con cuatro botones (con las etiquetas: manual, automático, abrir y cerrar), textos, una imagen con el logo de la universidad y una animación para indicar el nivel de iluminación, tal y como se muestra en la figura 23.



Figura 23: Interfaz gráfica de la aplicación móvil

Seguidamente se procedió a realizar la programación de la interfaz, considerando importar los elementos propuestos en la figura 23 y la información que se desea extraer de la base de datos, es decir las variables descritas en la tabla 4; además se aplican los logaritmos que describen las funciones de cada elemento de la aplicación y los datos que enviar a la base de dato en caso de que se active o desactive un botón específico; tal y como se muestra en el anexo 4.

6.2.4. Funcionamiento Integral del prototipo

El sistema operativo del prototipo está constituido de tres partes como se puede observar en la figura 24; en donde, la tarjeta enviará el nivel de iluminación que lee el sensor; esta información modificara los estados de las variables en la base de datos y a su vez serán leídos por la aplicación. De la misma forma, la interfaz de la aplicación móvil enviará datos que modifiquen los estados en la base de datos y a su vez estos serán leídos por la tarjeta ESP32 para ejecutar las acciones correspondientes al giro del motor e indicadores LED, tal y como se muestra en la 25; siempre y cuando en el dispositivo físico este habilitado la opción correspondiente.

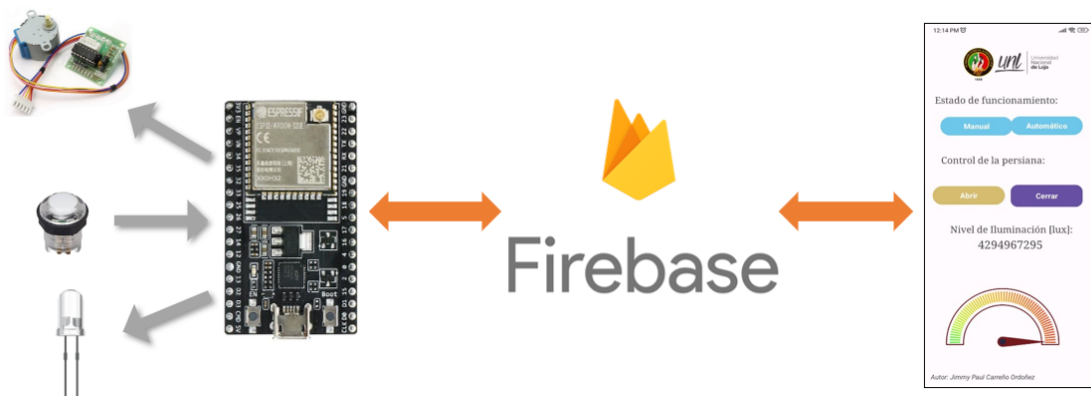


Figura 24: Comunicación Integral del prototipo



Figura 25: Prueba del funcionamiento del prototipo

6.3 Implementación del prototipo

6.3.1. Construcción del prototipo

Después de haber compilado de manera correcta los algoritmos de programación tanto para aplicación móvil, como para la tarjeta ESP32; se procedió a realizar la construcción del prototipo tomando en cuenta la lista de materiales de la tabla 6. En donde se puede observar que el costo de los materiales del prototipo es de \$291.1, pero tomando en cuenta que la persiana es el elemento más representativo económicamente, el costo del control de la persiana automática es de \$78.3.

Tabla 6: Materiales usados para la construcción del prototipo

Orden	Detalle	Cantidad	P.Unitario	P.Total
1	Módulo ESP-32 WROOM/Dev Module	1	15	15
2	BH1750	1	4	4
3	Cable UTP Cat.6	2m	0.5	1
4	Placa PCB perforada para soldar	2	0.70	1.40
5	Persiana 1.5mx4m	1	212.8	212.8
6	Botones azules	2	0.5	1
7	Caja para interruptor sobrepuesto	1	0.90	0.90
8	Fuente de 110VAC/5VDC	1	5	5
9	Rollo de Estaño	1	3	3
10	Tapa para cajetín	1	0.5	0.5
11	Pantalla LCD 16x2	1	8	8
12	Interruptor de dos vías	2	0.5	1
13	NEMA17	1	20	20
14	IC Driver controlador	1	8	8
15	Switch de dos vías	2	0.4	0.8
16	Tornillos con doble tuerca	4	0.05	0.2
17	Polea de grilon	1	5	5
Total:				297.1

Una vez recopilado todos los materiales necesarios para la construcción del prototipo, tal y como se muestra en la figura 26, se procede a realizar la soldadura de los componentes considerando el esquema de conexión del anexo 2, tal y como se muestra en la figura 27, para ello se consideró que el prototipo pueda ser desmontable y que el panel de control cuente con las aberturas necesarias para los actuadores.



Figura 26: Materiales que conforman el control del prototipo

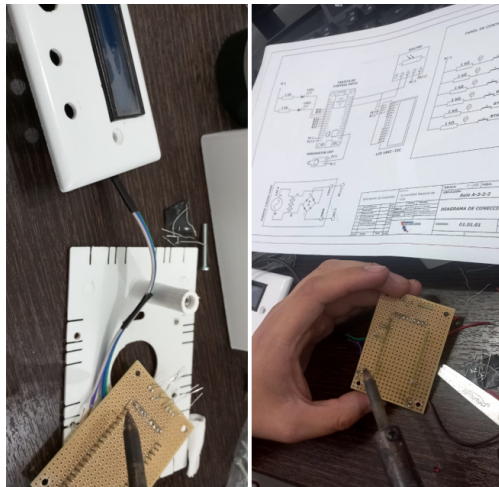


Figura 27: Soldadura del circuito

Seguidamente se procede a realizar un ensamble de todos los componentes, para ello se procuró construir un dispositivo que sea versátil, de fácil desmontaje y que circuito sea capaz de encajar en la caja de un interruptor sobrepuesto, tal y como se muestra en la figura 28.

A continuación se realiza la integración del motor a la persiana y luego se procede a instalar el dispositivo de control, de tal manera que las láminas se muevan para dar apertura o cierre de la luz natural y considerando las fuentes de alimentación del circuito, tal y como se muestra en la figura 29

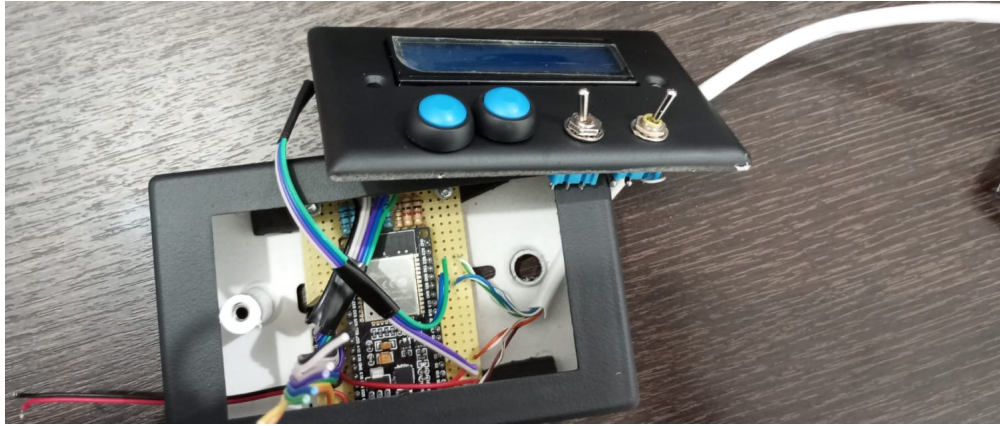


Figura 28: Ensamble del prototipo



Figura 29: Implementación del prototipo

6.3.2. Implementación del prototipo en la zona de emplazamiento

Una vez realizado las pruebas de funcionamiento del prototipo, en el código de programación de la tarjeta ESP32 se realizaron las modificaciones correspondientes al usuario y contraseña de la red wifi de la zona de emplazamiento, en los apartados de `<define ssid>` y `<define pass>`, por lo tanto, se compiló y cargó el código nuevamente.

Seguidamente se procedió a instalar la persiana en la ventana del aula A-3-2-2, para ello, se calculó el peso total de las aspas persianas de la persiana que tiene que hacer girar el motor, a continuación se realizó la medición de las dimensiones y peso de una sección de las aspas de una persiana; dando como resultado que la misma tiene una dimensión de 9.2 cm por 30.6 cm; con un peso de 80 g como se evidencia en la figura 30.

Por lo tanto, de acuerdo a la figura 30 para una longitud de 0.306 m la persiana tiene una masa de 0.08 kg y considerando que el alto de la ventana es de 1.5 m, el peso total de un aspa de la persiana es de 0.3922 kg. Además, con el ancho de la ventana (4 m) y el ancho de un aspa (0.092 m), se determina que el número total de aspas que tendrá la persiana, es decir:



Figura 30: Dimensiones y peso de una sección de las aletas de la persiana

$$N = \frac{4 \text{ m}}{0.092 \text{ m}} = 43.47 \approx 44$$

Por lo cuál, el peso total de las persianas es el producto del peso de un aspa (0.3922 kg) y el número de aspas (44), dando como resultado 17.26 kg. El sistema mecánico que controla el giro de las aspas de la persiana está conformado por una polea que mueve un eje dentado que a su vez generan el movimiento de las aspas con un engranaje cónico como se observa en la figura 31; en donde, dependiendo del tipo de motor, marca y capacidad de carga, este puede girar a diferentes revoluciones por minuto como se observa en la tabla 7.

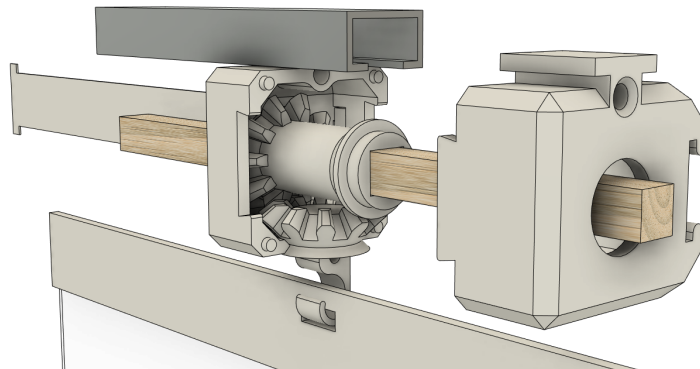


Figura 31: Mecanismo de movimiento de la persiana vertical

Fuente: (FLOL, 2022).

Seguidamente, se procedió a realizar el cálculo del torque necesario para girar las aspas de la persiana de acuerdo a la **Ecuación (10)**, en donde, se tomó en cuenta que no existe

Tabla 7: Características de motores de persianas disponibles en el mercado

Denominación	Capacidad de carga [kg]	Velocidad angular [rpm]
AM-45 BASIC	32	17
Jarofilt SL 10/17	25	17
Motor marca Coulisse	25	4
MAA-YH002	10	45
ES35-10	25	15
DM35R-10	33	14
Promedio		18.67 \approx 19

resistencia del viento para que la aspa gire, ya que el la persiana se encuentra en un lugar cerrado y no expuesto a la interperie, por lo que se la considera como un cuerpo de cilindro sólido, cuyo peso es de 0,3922 kg; el radio de una aspa que de acuerdo a la figura 30 es $0,092/2$ m=0.046 m y según la tabla 7 las revoluciones promedio de las persianas es de $19\text{rpm} = 2 \frac{\text{rad}}{\text{s}}$, por lo que por cada segundo se tendría una aceleración angular de $2 \frac{\text{rad}}{\text{s}^2}$:

$$T = \frac{1}{2}mr^2\alpha = \frac{1}{2}(0.3922 \text{ kg})(0.046 \text{ m})^2(2 \frac{\text{rad}}{\text{s}^2}) = 8.3 * 10^{-4} [\text{Nm}]$$

Por lo tanto, al multiplicar el número de aspas (44) por el torque requerido por cada aspa ($8.3 * 10^{-4}[\text{Nm}]$), se obtiene que el sistema requiere un torque total de 0.03652Nm . Por lo cual, se selección un motor paso a paso disponible en el mercado local que cumpla con los requerimientos del sistema; siendo el mismo, el NEMA17 17HS4401, cuyo torque es de 0.4 Nm de acuerdo a las características técnicas del datashet del anexo 6. Al cuál se lo instala junto al equipo de control de la persiana, con su respectiva polea para la cadena, cuyo plano se evidencia en el anexo 2.

Por último, se comprobó que el prototipo se encuentre debidamente conectado a una fuente de energía y se realizan pruebas de funcionamiento como se observa en la figura 32.

6.3.3. Calibración

Seguidamente, se realizó la medición de los niveles de iluminación interior y exterior del aula con varios luxómetros BH1750 al mismo tiempo, siguiendo el protocolo de la normativa UNE-EN 12464-1, para identificar la incidencia de la persiana en el paso de luz natural dentro del aula y que proporción se aprovecha de esta; obteniendo los resultados que se muestran en la figura 33.

De acuerdo a la figura 33 se obtuvieron diferentes niveles de iluminación a nivel in-



Figura 32: Instalación y prueba de funcionamiento del prototipo

terior y exterior. En la parte exterior se obtuvieron mediciones que van desde los 3647 lx hasta los 10973 lx, por lo cual, el nivel de iluminación que se obtuvo después de la persiana fue de 1623 lx y 2728 lx respectivamente, estando por encima de la norma, sin embargo, las mediciones realizadas a nivel medio de aula como se evidencia en los indicadores naranjas de la gráfica, se obtuvieron niveles de iluminación de 195 lx y 535 lx respectivamente; por lo tanto al trazar la líneas de tendencia de ambas mediciones se identificó que para garantizar un nivel de iluminación óptimo a nivel medio de la superficie total para el aula de un centro académico se debe contar con un nivel de iluminación exterior de 4300 lx y 10900 lx aproximadamente; debido a que si se obtiene nivel de iluminación inferiores, no se garantiza una buena iluminación dentro del aula y si se obtiene niveles de iluminación exterior superiores a eso, se obtiene un excedente de iluminación a nivel interior. Además la línea de tendencia que define la iluminación exterior (ILI) con respecto a la iluminación exterior (ILE) es la siguiente:

$$ILI = 0.0367 * ILE + 90.529 \text{ [lx]}$$

Por lo tanto, para garantizar el correcto funcionamiento del prototipo en forma automática, se estableció que si los niveles de iluminación exterior son inferiores a los 4300 lx la persiana opte por cerrar completamente para que quede a consideración de los usuarios el utilizar la iluminación artificial. De la misma forma en el caso que los niveles de iluminación exterior se encuentren dentro de los 4300 lx a 10900 lx, la persiana se abra completamente con la finalidad de aprovechar al máximo la iluminación natural. Por último, en el caso que los niveles de iluminación exterior excedan los límites establecidos la persiana se cerrará parcialmente considerando que el motor paso a paso solo requiere dar 5 giros completos para cerrar la persiana; es decir por cada 500 lx de iluminación excedente que se tenga, la persiana se

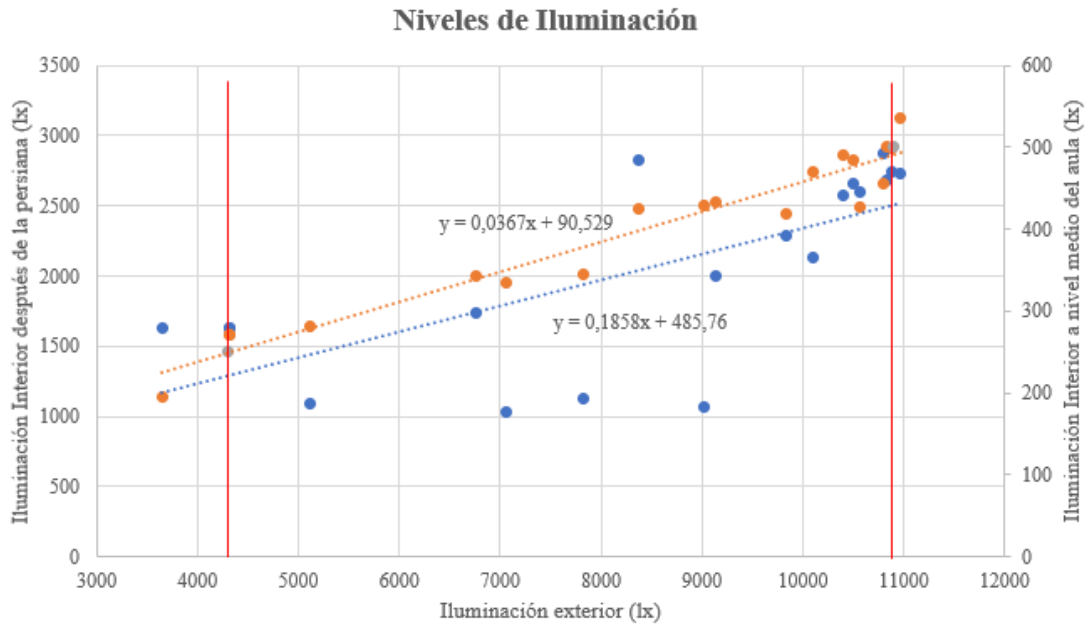


Figura 33: Niveles de medición obtenidos para calibración

cerrará un giro; esto con la finalidad de obtener un rango de aprovechamiento de iluminación y porque no se puede conocer con exactitud cuál es el límite máximo de iluminación exterior que se puede obtener para la zona de emplazamiento.

7. Discusión

Con la colaboración del luxómetro BH1750 y mediante una práctica experimental, se identificaron los niveles de iluminación del aula A-3-2-2 del edificio 3 de la Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables; en donde, según la normativa UNE-EN 12464-1, se determinó que el Factor de Luz Diurna (FLD) se encuentra dentro del rango permisible y en la gráfica de calor se identificaron niveles de iluminación superiores a los 500 lx requeridos por las regulaciones.

Mediante uso la tarjeta ESP32, la base de datos de Firebase y Android Studio, se diseñó un prototipo de persianas automáticas capaz de regular el nivel de iluminación natural en las aulas; en donde, el sistema sistema puede ser controlado físicamente o a través de una aplicación móvil y ofrece modos de funcionamiento manual y automático; para ello, se diseñó una interfaz gráfica y un panel de control, que muestran los niveles de iluminación exterior y proporciona acceso a las diversas funciones; por lo tanto, este diseño tiene la ventaja de ser precursor en el desarrollo de sistemas de gestión energética en las aulas de una Universidad Nacional de Loja con tecnología IoT y comunicación con bases de datos en la red.

Con los componente disponibles en el mercado local, se construyó el prototipo diseñado, con un costo de inversión del mecanismo de control de \$ 78.30, el cual resulta competitivo ante otros existentes en el mercado internacional que carecen de las mismas características del prototipo propuesto, además, al ofrecer un costo de operación más bajo en comparación con el uso de iluminación artificial, el prototipo se presenta económicamente viable; cuyo funcionamiento fue debidamente validado en la zona de estudio, incluyendo la calibración del mismo con los niveles de iluminación tanto exterior como interior.

8. Conclusiones

- Con la ayuda del luxómetro BH1750 y mediante una práctica experimental, se identificaron los niveles de iluminación del aula A-3-2-2 del bloque 3 de la Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables; en donde, de acuerdo con la UNE-EN 12464-1, el FLD está dentro del rango permisible y en la gráfica de calor se identificó niveles de iluminación mayores a los 500 lx requeridos por las normativas.
- Con la tarjeta ESP32, la base de datos de Firebase y Android Studio, se diseñó un prototipo de persianas automáticas que regulen el nivel de iluminación en las aulas; cuyo funcionamiento tiene una modalidad de control en físico o mediante aplicativo móvil, con un estado de funcionamiento manual y automático; para ello se diseñó una interfaz gráfica y un panel de control, que muestra los niveles de iluminación exterior y los mandos que accedan a cada función.
- Con los componente disponibles en el mercado local, se construyó el prototipo diseñado cuyo costo de inversión es de \$291.1 y se realizó la validación de su correcto funcionamiento al ser implementado en la zona de estudio, cubriendo la necesidad de control de iluminación exterior y realizando la calibración del mismo considerando el número de pasos del motor y el nivel de iluminación interior requerido con respecto al exterior.
- En la presente investigación se implementó un prototipo de persianas automáticas que permitan regular el nivel de luminosidad en las aulas de FEIRNNR, para ello se hizo uso de tecnología IoT y lectura de datos en una base de datos de la red, al dispositivo se lo puede controlar mediante una aplicación móvil a través de internet y mediante un panel de control físico; además, descartando el costo de la persiana, el precio del mecanismo de control es de \$78.30, siendo este competitivo con otros existentes en el mercado internacional que no cuentan con las mismas características que el prototipo propuesto.

9. Recomendaciones

- Mediante la metodología planteada, identificar los niveles de iluminación de las aulas de otros bloques de la la Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables; de la misma forma, considerar las otras variables de ergonomía que requieren las personas que ejecutan actividades académicas, como la ventilación y control de temperatura para tener un prototipo domótico que mejore las condiciones ambientales de un aula de clase.
- Al cumplir con los objetivos propuestos, se propone implementar mejoras el diseño del prototipo de control, con la finalidad de integrar un sistema de comunicación de doble vía, que facilite el control de la persiana con los dos mecanismos al mismo tiempo. De la misma forma, implementar un módulo que permita el control con asistencia de voz como alexa e integrar la domotización completa del aula.
- Implementar estrategias que reduzcan el costo de producción del prototipo para que sea asequible para un mayor público.
- Por el alcance del proyecto, se propone implementar el prototipo en la mayoría de las aulas de la FEIRNNR, con la finalidad de gestionar de manera adecuada la energía en la Facultad.

10. Bibliografía

- Alberto Perez, P., Perez Hidalgo, A., & Perez Berenguer, E. (2017). *Introducción a los sistemas de control y modelo matemático para sistemas lineales invariantes en el tiempo* (Tesis de grado). Universidad Nacional de San Juan. España. <http://dea.unsj.edu.ar/control1/apuntes/unidad1y2.pdf>
- Android. (2023). Android para desarrolladores. <https://developer.android.com/studio/intro?hl=es-419>
- Babiuch, M., Foltynek, P., & Smutny, P. (2019). Uso del microcontrolador ESP32 para el procesamiento de datos. *IEEE*, 18. <https://doi.org/10.1109/CarpathianCC.2019.8765944>
- Bertoletti, P. (2019). *Proyectos con SP32 y LoRa* (Primera). Newton C Braga. https://books.google.com.ec/books?hl=es&lr=&id=Doi0DwAAQBAJ&oi=fnd&pg=PT4&dq=esp+32+articulo+cientifico&ots=mBpqFhvJol&sig=X1wAemS24TG_0eS9rlAL6bqKiJQ#v=onepage&q=esp%2032%20articulo%20cientifico&f=false
- Bourhis, P., Reutter, J. L., & Vrgoč, D. (2020). JSON: Data model and query languages. *Information Systems*, 89, 101478.
- Cámara, J. I. (2020). *Diseño, realización y control de un robot autoequilibrado de bajo coste basado en una Raspberry PI* (Trabajo Fin de Grado). Escuela Técnica Superior de Ingeniería. Sevilla. <https://idus.us.es/bitstream/handle/11441/102314/TFG-2965-CAMARA%20MOLINA.pdf?sequence=1&isAllowed=y>
- Conti, F. (2005). *Motores paso a paso* (Segunda, Vol. 2). ALSINA. https://books.google.es/books?hl=es&lr=&id=6yiXDwAAQBAJ&oi=fnd&pg=PA1&dq=motor+paso+a+paso&ots=_YVvdXReus&sig=ud1uQ_iivnLC037GHH4T0nLDv08#v=onepage&q=motor%20paso%20a%20paso&f=false
- Crespo, E. (2016). IDE Arduino. Consultado el 25 de abril de 2023, desde <https://aprendiendoarduino.wordpress.com/2016/12/11/ide-arduino/>
- Cruz Vega, M., Oliete Vivas, P., Morales Ríos, C., Gonzáles Luis, C., Cendón Martín, B., & Hernández Seco, A. (2015). *Las tecnologías IoT dentro de la industria conectada*

- 4.0 (Primera, Vol. 3). Funadación EOI. <https://books.google.com.ec/books?hl=es&lr=&id=ujRaDwAAQBAJ&oi=fnd&pg=PA4&dq=tecnologia+iot+articulo+cientifico&ots=Wgie1utQPP&sig=ZsjkCn167UEBuCfLLd9EJgElB8Q#v=onepage&q=tecnologia%20iot%20articulo%20cientifico&f=false>
- Cué, J. A. (2018). *Diseño e implementación de una interfaz electromecánica para el manejo de sillas de ruedas motorizadas* (Trabajo Fin de Grado). Universidad Politécnica de Madrid. Madrid. https://oa.upm.es/49668/1/TFG_JOSE_ANTONIO_CUE_PALERO.pdf
- Díaz, J. (2021). Desarrollo de una herramienta digital con comunicación a tiempo real usando Firebase para la automatización de procesos en Krosmaster, un juego de mesa.
- FLOL, F. (2022). Vertical blinds mechanism. Consultado el 17 de julio de 2023, desde <https://www.printables.com/es/model/177067-vertical-blinds-mechanism>
- INEN-1.152. (1984). *ILUMINACIÓN NATURAL DE EDIFICIOS. REQUISITOS* (Vol. 4). INEN. https://www.normalizacion.gob.ec/buzon/normas/nte_inen_1152.pdf
- Maps, G. (2023). Ubicación de la Universidad Nacional de Loja. Consultado el 15 de junio de 2023, desde https://www.google.com.ec/maps/place/Universidad+Nacional+de+Loja/@-4.0331644,-79.2035347,16z/data=!4m6!3m5!1s0x91cb3781a18539a9:0x42937e413fcf09eb!8m2!3d-4.0329396!4d-79.2025477!16s%2Fg%2F1hb_fw124?entry=ttu
- Martínez Vaca, D. A. (2021). *Estudio comparativo de las mejoras del lenguaje de programación kotlin y el lenguaje java en el desarrollo de aplicaciones android* (B.S. thesis). BABAHOYO: UTB, 2021.
- Monroy, M. M. (2014). *Manual de la Iluminación* (Primera, Vol. 1). ICARO. <https://m2db.files.wordpress.com/2014/09/manual-1-iluminacion.pdf>
- Mossi, J. M. (2016). *FOCUS ASSISTANT: SISTEMA REMOTO DE CONTROL DE FOCO* (Trabajo Fin de Grado). Universitat Politècnica de Valencia. Valencia. <https://riunet.upv.es/bitstream/handle/10251/80466/Ja%c3%a9n%20-%20Focus%20assistant.%20Sistema%20remoto%20de%20control%20de%20foco.%20Unidad%20actuadora.pdf?sequence=1&isAllowed=y>
- Norma-Ecuatoriana-de-la-Construcción. (2018). *Eficiencia Energética en edificaciones residenciales (EE)* (Vol. 4). Ministro de Desarrollo Urbano y Vivienda. <https://www.habitatyvivienda.gob.ec/wp-content/uploads/2023/03/4.-NEC-HS-Eficiencia-Energetica.pdf>
- Ordóñez, A. (2021). Método de análisis estático. Consultado el 18 de abril de 2023, desde <https://www.seiscubos.com/conocimiento/metodos-de-analisis-estaticos>

- Pucha, D. F. (2023). *Diseño y construcción de un prototipo de lámpara inteligente para el control de iluminación led mediante tecnología inalámbrica Lora* (Tesis de grado). Universidad Nacional de Loja. Loja. https://dspace.unl.edu.ec/jspui/bitstream/123456789/26055/1/DiegoFernando_%20PuchaCaraguay.pdf
- Robledo, D. (2018). *Desarrollo de aplicaciones para Android I* (1.^a ed., Vol. 1). Aula mentor. <https://sede.educacion.gob.es/publiventa/PdfServlet?pdf=VP18117.pdf&area=E>
- Roman Arenaza, R. E. (2019). *Lenguajes de programación Javascript Java y Javascript. Características. Norma de escritura. Variables y operadores lógicos. Mensajes. Ejercicios. Estructuras condicionales. Funciones y objetos. Aplicaciones.*
- Samsung. (2023). *Curso UMA / Samsung DESARROLLO DE APPS PARA ANDROID* (Samsung TECH INSTITUTE, Vol. 2). https://www.uma.es/media/tinyimages/file/android_ed2.pdf
- Sánchez, L. H. (2022). *Sistema de iluminación LED automatizada para el edificio 3 de la Facultad de Energía, las Industrias y los Recursos Naturales no Renovables de la Universidad Nacional de Loja* (Trabajo de Titulación). Universidad Nacional de Loja. Loja. Consultado el 2 de marzo de 2022, desde <https://dspace.unl.edu.ec/jspui/handle/123456789/25972>
- Silveira Amaro, D. (2020). *Automatización y control de persianas eléctricas mediante micro-controladores* (Tesis de grado). UNIVERSIDAD CARLOS III DE MADRID. España. <https://e-archivo.uc3m.es/handle/10016/32780#preview>
- Slüsarenko, V., & Rojas, R. (2021). Uso de cuplas o pares de fuerzas para expresar el torque en situaciones de equilibrio estático. *43*(43). <https://doi.org/https://doi.org/10.1590/1806-9126-RBEF-2020-0476>
- Solano, N. C. (2022). *Exploración de los problemas de seguridad que presenta el IoT en el contexto del Edge Computing* (Tesis de grado). Universidad Nacional de Loja. Loja. Consultado el 19 de abril de 2023, desde https://dspace.unl.edu.ec/jspui/bitstream/123456789/24789/1/NoraCecivel_SolanoChamba..pdf
- Sylvania. (2021). *Manual técnico de iluminación* (Segunda, Vol. 2). <https://sylvania.com.ec/wp-content/uploads/2021/01/Manual-t%C3%A9cnico-de-iluminaci%C3%B3n-Sylvania.pdf>
- Taboada, J. A. (1983). *Manual de luminotecnia* (Cuarta, Vol. Cuarta). OSRAM.
- Tejeda y Azpeitia, G. (2015). *Prontuario solar de México* (Primera, Vol. 1). enfoque académico. https://www.researchgate.net/profile/Luis-Gabriel-Gomez-Azpeitia/publication/312576876_Prontuario_Solar_de_Mexico/links/58823f144585150dde401d19/Prontuario-Solar-de-Mexico.pdf#page=70

UNE-12464. (2022). *Luz e iluminación. Iluminación de los lugares de trabajo. Parte 1: Lugares de trabajo en interiores* (Última, Vol. 1). Normalización Española. <https://es.scribd.com/document/602068300/UNE-12464#>

Villalón Pardo, D. (2021). *Crear y desarrollar una aplicación de alto rendimiento con bajo coste utilizando flutter y firebase* (Tesis doctoral). Universitat Politècnica de València.

11. Anexos

Anexo I: Práctica experimental



unl

Universidad
Nacional
de Loja

Diseño y construcción de un prototipo de persiana con ajuste automático de iluminación para las aulas de bloque 3 de la FEIRNNR

MEDICIÓN DE LOS NIVELES DE ILUMINACIÓN DEL AULA A-3-2-2

INTRODUCCIÓN

Con la finalidad de conocer los niveles de iluminación que existen en la zona de emplazamiento de la investigación, se procedió a llevar a cabo la siguiente práctica experimental en la Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables, con la finalidad de obtener los parámetros de calibración del prototipo.

OBJETIVO

Obtener los niveles de iluminación del aula A-3-2-2 mediante el uso del luxómetro BH1750.

MATERIALES

-) Dos luxómetros BH1750
-) Tarjeta Arduino Uno.
-) Insumos de oficina.
-) Computadora.
-) Paquete de Microsoft Office.
-) Flexómetro.
-) Cinta masking.
-) Ovillo de hilo.

PROCEDIMIENTO

1. Dirigirse a la zona de emplazamiento y limpiar la ventana que permite el acceso de la luz natural.
2. Conectar los dos sensores a la placa de control y realizar la programación, considerando instalar las librerías para el sensor de BH1750.
3. Realizar la programación de la tarjeta de Arduino Uno, con el siguiente algoritmo:

```
1. //Importar librerías
2. #include <BH1750.h>
3. #include <Wire.h>
4.
5. BH1750 sensor1; //asiganamos los datos de escritura del sensor
6. BH1750 sensor2; //asiganamos los datos de escritura del sensor
7. //Definimos los puntos de control
8. const int L=9; //BOTON PARA MEDIR
9.
10. void setup() {
11. // put your setup code here, to run once:
12. Wire.begin();
```

```
13. sensor1.begin();
14. sensor2.begin();
15. pinMode(L,INPUT);
16. Serial.begin(9600);
17.}
18.
19.void loop() {
20. // put your main code here, to run repeatedly:
21. if(digitalRead(L)==HIGH){
22.   unsigned int lux1=sensor1.readLightLevel();
23.   Serial.print("Nivel interior: ");
24.   Serial.print(lux1);
25.   Serial.println(" lx");
26.   delay(5000);
27.
28.   unsigned int lux2=sensor2.readLightLevel();
29.   Serial.print("Nivel exterior: ");
30.   Serial.print(lux2);
31.   Serial.println(" lx");
32.   delay(5000);
33. }
34.}
```

4. Construir el prototipo de medición tomando en cuenta el diagrama que se muestra en la Figura 1.

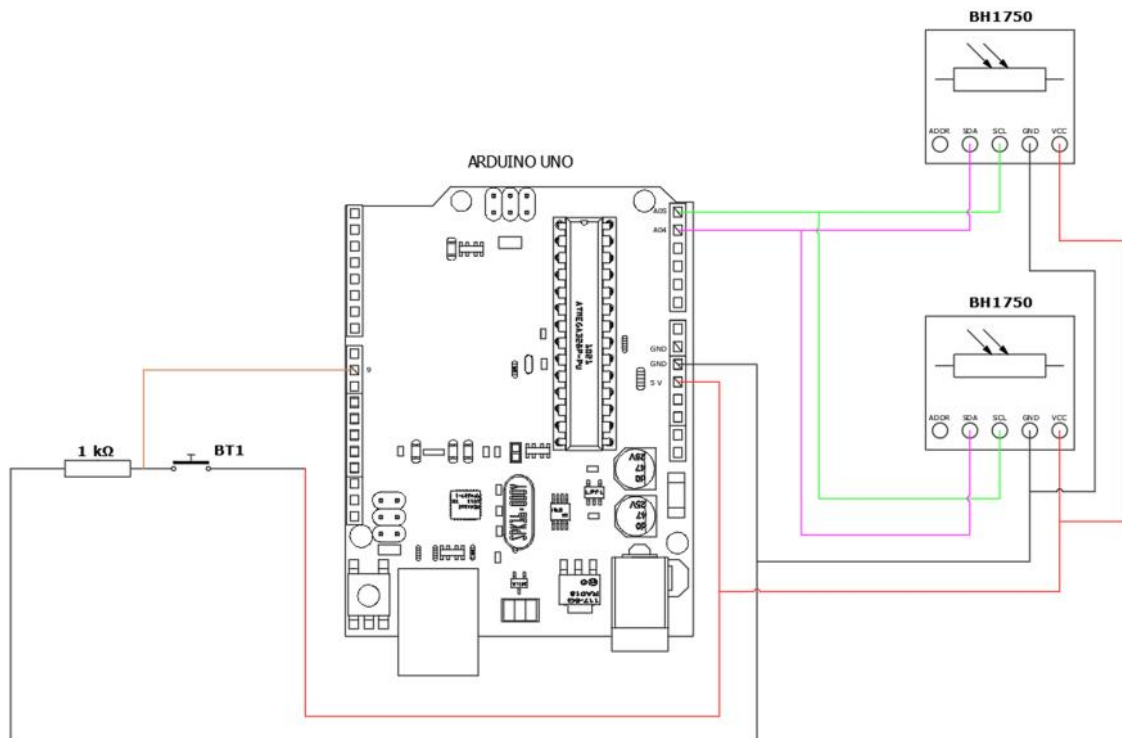


Figura 1. Diagrama de conexión del instrumento de medición

- En la zona de emplazamiento, con la ayuda del flexómetro, cinta masking y el ovillo de hilo, se procede a marcar la malla para los puntos de medición en el salón de clases, tal y como se muestra en la Figura 2.

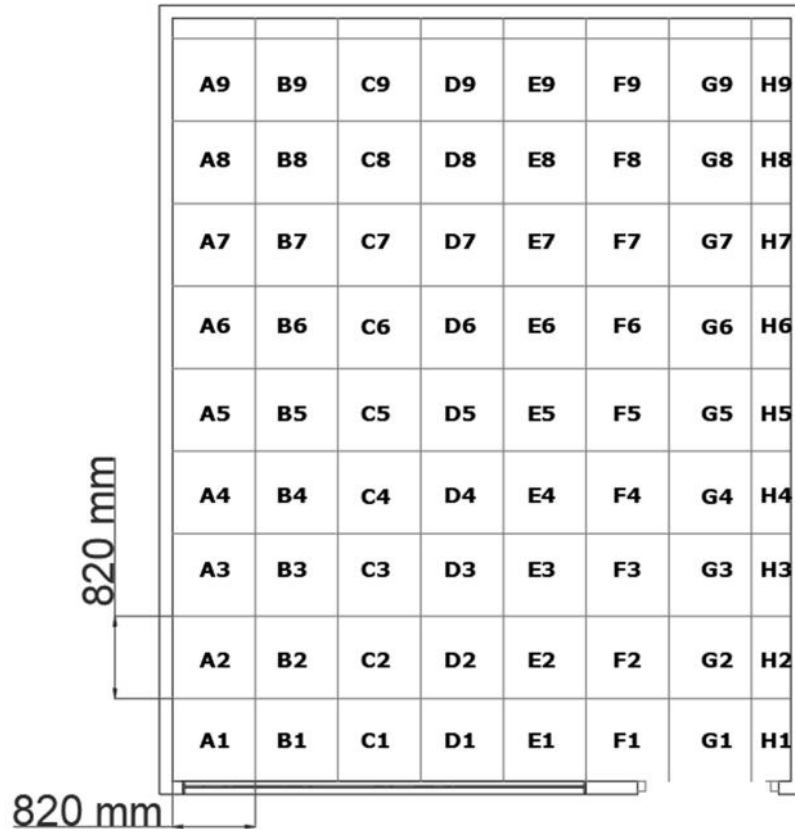


Figura 2. Malla de la zona de medición

- Despejar el salón de clases para realizar las mediciones en la altura que los estudiantes requieren para realizar sus actividades.
- Realizar el levantamiento de la información con puerto serial del IDE de Arduino y registrar los datos en Excel con el formato de la Tabla 1; considerando que se realizaran 5 mediciones por cada malla cada 5 segundos, a la altura de la mesa de los pupitres, debido a que es la principal zona de incidencia de la luz natural.

Tabla 1. Información referencial para tabular datos

Código de Malla	Tiempo	Nivel de iluminación exterior (lx)	Nivel de iluminación interior (lx)
PROMEDIO:			

Fuente: El autor



UNL

Universidad Nacional de Loja

Diseño y construcción de un prototipo de persiana con ajuste automático de iluminación para las aulas de bloque 3 de la FEIRNNR

RESULTADOS

Los niveles de iluminación obtenidos fueron los siguientes:

Tabla 2. Mediciones realizadas

Código de Malla	Tiempo	Nivel de iluminación interior (lx)	Nivel de iluminación exterior (lx)
A1	12:00:00	1180	38715
A1	12:00:05	1290	38166
A1	12:00:10	1274	38395
A1	12:00:15	1280	37920
A1	12:00:20	1303	38074
Promedio A1		1265,40	38254,00
B1	12:00:25	3702	38134
B1	12:00:30	3666	38377
B1	12:00:35	3521	38170
B1	12:00:40	3517	40489
B1	12:00:45	3554	41921
Promedio B1		3592,00	39418,20
C1	12:00:50	3557	41834
C1	12:00:55	3568	41214
C1	12:01:00	1485	40106
C1	12:01:05	1424	39665
C1	12:01:10	1382	39845
Promedio C1		2283,20	40532,80
D1	12:01:15	1470	44194
D1	12:01:20	1544	41750
D1	12:01:25	3589	47396
D1	12:01:30	3608	28992
D1	12:01:35	3585	43496
Promedio D1		2759,20	41165,60
E1	12:01:40	3585	43525
E1	12:01:45	3608	48036
E1	12:01:50	3589	49361
E1	12:01:55	1544	44275
E1	12:02:00	1303	39515
Promedio E1		2725,80	44942,40
F1	12:02:05	3702	38766
F1	12:02:10	3666	47446
F1	12:02:15	3521	44117
F1	12:02:20	3517	41410
F1	12:02:25	3554	42544
Promedio F1		3592,00	42856,60
G1	12:02:30	3557	43461
G1	12:02:35	3568	49861
G1	12:02:40	1485	46099
G1	12:02:45	1424	44057
G1	12:02:50	1382	38463
Promedio G1		2283,20	44388,20
H1	12:02:55	1470	40307
H1	12:03:00	1280	42741
H1	12:03:05	1274	42501
H1	12:03:10	1290	38789
H1	12:03:15	1180	38307
Promedio H1		1298,80	40529,00
A2	12:03:20	3760	44128



UNL

Universidad
Nacional
de Loja

Diseño y construcción de un prototipo de persiana con ajuste automático de iluminación para las aulas de bloque 3 de la FEIRNNR

A2	12:03:25	3867	44046
A2	12:03:30	4127	40742
A2	12:03:35	4426	40206
A2	12:03:40	4487	43722
Promedio A2		4133,40	42568,80
B2	12:03:45	3906	40165
B2	12:03:50	2846	45481
B2	12:03:55	2856	43640
B2	12:04:00	2835	43687
B2	12:04:05	2791	38545
Promedio B2		3046,80	42303,60
C2	12:04:10	2387	41905
C2	12:04:15	2377	47338
C2	12:04:20	4480	47990
C2	12:04:25	4565	42422
C2	12:04:30	4474	46526
Promedio C2		3656,60	45236,20
D2	12:04:35	4277	42774
D2	12:04:40	4075	42411
D2	12:04:45	2758	44993
D2	12:04:50	2710	44481
D2	12:04:55	2672	38707
Promedio D2		3298,40	42673,20
E2	12:05:00	3760	46407
E2	12:05:05	3867	40791
E2	12:05:10	4127	40667
E2	12:05:15	4426	44434
E2	12:05:20	4487	39736
Promedio E2		4133,40	42407,00
F2	12:05:25	3906	39637
F2	12:05:30	2846	48165
F2	12:05:35	2856	40418
F2	12:05:40	2835	41674
F2	12:05:45	2791	41525
Promedio F2		3046,80	42283,80
G2	12:05:50	2387	38731
G2	12:05:55	2377	47517
G2	12:06:00	4480	44005
G2	12:06:05	4565	48746
G2	12:06:10	4474	39596
Promedio G2		3656,60	43719,00
H2	12:06:15	4277	41038
H2	12:06:20	4075	42380
H2	12:06:25	2758	42253
H2	12:06:30	2710	41085
H2	12:06:35	2672	44328
Promedio H2		3298,40	42216,80
A3	12:06:40	2375	39642
A3	12:06:45	2383	49855
A3	12:06:50	2415	43947
A3	12:06:55	2463	43381
A3	12:07:00	2461	45284
Promedio A3		2419,40	44421,80
B3	12:07:05	2835	39115
B3	12:07:10	2791	45838
B3	12:07:15	2758	46416
B3	12:07:20	2710	46797



UNL

Universidad
Nacional
de Loja

Diseño y construcción de un prototipo de persiana con ajuste automático de iluminación para las aulas de bloque 3 de la FEIRNNR

B3	12:07:25	2342	39834
Promedio B3		2687,20	43600,00
C3	12:07:30	1696	43339
C3	12:07:35	1689	41858
C3	12:07:40	1670	44115
C3	12:07:45	2032	46413
C3	12:07:50	2470	40230
Promedio C3		1911,40	43191,00
D3	12:07:55	2500	41192
D3	12:08:00	2462	38047
D3	12:08:05	2413	40966
D3	12:08:10	2370	45373
D3	12:08:15	2758	45886
Promedio D3		2500,60	42292,80
E3	12:08:20	2710	47285
E3	12:08:25	2758	47835
E3	12:08:30	2710	41845
E3	12:08:35	2387	38220
E3	12:08:40	2377	49922
Promedio E3		2588,40	45021,40
F3	12:08:45	2375	44366
F3	12:08:50	2383	46690
F3	12:08:55	2415	39019
F3	12:09:00	2463	43485
F3	12:09:05	2461	40287
Promedio F3		2419,40	42769,40
G3	12:09:10	2470	40641
G3	12:09:15	2500	49856
G3	12:09:20	2462	44384
G3	12:09:25	2413	46115
G3	12:09:30	2370	43453
Promedio G3		2443,00	44889,80
H3	12:09:35	2342	39882
H3	12:09:40	2500	48910
H3	12:09:45	2462	38573
H3	12:09:50	2413	46078
H3	12:09:55	2710	39931
Promedio H3		2485,40	42674,80
A4	12:10:00	1696	39830
A4	12:10:05	1689	45723
A4	12:10:10	1670	38240
A4	12:10:15	2032	48419
A4	12:10:20	1949	49299
Promedio A4		1807,20	44302,20
B4	12:10:25	1924	41061
B4	12:10:30	1924	42577
B4	12:10:35	1919	44748
B4	12:10:40	1922	39974
B4	12:10:45	1189	42409
Promedio B4		1775,60	42153,80
C4	12:10:50	1204	38127
C4	12:10:55	1180	40457
C4	12:11:00	1219	40626
C4	12:11:05	1125	47821
C4	12:11:10	1210	42147
Promedio C4		1187,60	41835,60
D4	12:11:15	1218	48296



UNL

Universidad
Nacional
de Loja

Diseño y construcción de un prototipo de persiana con ajuste automático de iluminación para las aulas de bloque 3 de la FEIRNNR

D4	12:11:20	1221	48246
D4	12:11:25	1226	41628
D4	12:11:30	1226	44281
D4	12:11:35	1235	47716
Promedio D4		1225,20	46033,40
E4	12:11:40	1949	40880
E4	12:11:45	1924	47910
E4	12:11:50	1924	38654
E4	12:11:55	1919	42039
E4	12:12:00	1922	40584
Promedio E4		1927,60	42013,40
F4	12:12:05	1670	49129
F4	12:12:10	2032	41146
F4	12:12:15	1949	48029
F4	12:12:20	1924	39735
F4	12:12:25	1924	43134
Promedio F4		1899,80	44234,60
G4	12:12:30	1919	40670
G4	12:12:35	1922	43787
G4	12:12:40	1189	44783
G4	12:12:45	1204	49330
G4	12:12:50	1180	44248
Promedio G4		1482,80	44563,60
H4	12:12:55	1219	46238
H4	12:13:00	1125	39739
H4	12:13:05	1210	48029
H4	12:13:10	1218	45601
H4	12:13:15	1922	41933
Promedio H4		1338,80	44308,00
A5	12:13:20	1189	41178
A5	12:13:25	1204	44244
A5	12:13:30	1180	38587
A5	12:13:35	1219	42567
A5	12:13:40	1125	40265
Promedio A5		1183,40	41368,20
B5	12:13:45	1210	40384
B5	12:13:50	1218	39456
B5	12:13:55	1221	49308
B5	12:14:00	1226	39247
B5	12:14:05	1226	41612
Promedio B5		1220,20	42001,40
C5	12:14:10	1235	41961
C5	12:14:15	1223	44705
C5	12:14:20	1272	41593
C5	12:14:25	1297	48503
C5	12:14:30	1306	46850
Promedio C5		1266,60	44722,40
D5	12:14:35	1223	44631
D5	12:14:40	1272	39963
D5	12:14:45	1297	48351
D5	12:14:50	1306	40137
D5	12:14:55	1312	44640
Promedio D5		1282,00	43544,40
E5	12:15:00	1314	41134
E5	12:15:05	1315	49602
E5	12:15:10	1324	42457
E5	12:15:15	1322	48757



UNL

Universidad
Nacional
de Loja

Diseño y construcción de un prototipo de persiana con ajuste automático de iluminación para las aulas de bloque 3 de la FEIRNNR

E5	12:15:20	1325	43653
Promedio E5		1320,00	45120,60
F5	12:15:25	1341	49492
F5	12:15:30	1352	41061
F5	12:15:35	1356	49624
F5	12:15:40	1334	39968
F5	12:15:45	1270	43479
Promedio F5		1330,60	44724,80
G5	12:15:50	1200	40094
G5	12:15:55	1182	49179
G5	12:16:00	1190	38327
G5	12:16:05	1315	38329
G5	12:16:10	1324	43610
Promedio G5		1242,20	41907,80
H5	12:16:15	1341	49996
H5	12:16:20	1352	48116
H5	12:16:25	1356	45493
H5	12:16:30	1334	48664
H5	12:16:35	1223	42640
Promedio H5		1321,20	46981,80
A6	12:16:40	1312	44113
A6	12:16:45	1314	38408
A6	12:16:50	1315	38643
A6	12:16:55	1324	40554
A6	12:17:00	1322	44495
Promedio A6		1317,40	41242,60
B6	12:17:05	1356	45838
B6	12:17:10	1325	49424
B6	12:17:15	1341	41202
B6	12:17:20	1352	39378
B6	12:17:25	1356	47915
Promedio B6		1346,00	44751,40
C6	12:17:30	1334	43131
C6	12:17:35	1270	46136
C6	12:17:40	1200	45982
C6	12:17:45	1182	45756
C6	12:17:50	1190	49557
Promedio C6		1235,20	46112,40
D6	12:17:55	1182	49152
D6	12:18:00	908	41059
D6	12:18:05	933	43187
D6	12:18:10	872	42387
D6	12:18:15	938	42119
Promedio D6		966,60	43580,80
E6	12:18:20	874	43397
E6	12:18:25	857	40601
E6	12:18:30	868	40405
E6	12:18:35	918	40706
E6	12:18:40	870	46083
Promedio E6		877,40	42238,40
F6	12:18:45	921	46912
F6	12:18:50	936	43515
F6	12:18:55	896	48587
F6	12:19:00	864	49662
F6	12:19:05	944	47489
Promedio F6		912,20	47233,00
G6	12:19:10	911	45109



UNL

Universidad
Nacional
de Loja

Diseño y construcción de un prototipo de persiana con ajuste automático de iluminación para las aulas de bloque 3 de la FEIRNNR

G6	12:19:15	916	41506
G6	12:19:20	878	47169
G6	12:19:25	876	48684
G6	12:19:30	854	45204
Promedio G6		887,00	45534,40
H6	12:19:35	869	46880
H6	12:19:40	871	43731
H6	12:19:45	850	43665
H6	12:19:50	919	48350
H6	12:19:55	891	38270
Promedio H6		880,00	44179,20
A7	12:20:00	863	48920
A7	12:20:05	829	45066
A7	12:20:10	797	41211
A7	12:20:15	803	44685
A7	12:20:20	836	44911
Promedio A7		825,60	44958,60
B7	12:20:25	839	42207
B7	12:20:30	846	49039
B7	12:20:35	854	39954
B7	12:20:40	853	40990
B7	12:20:45	833	42529
Promedio B7		845,00	42943,80
C7	12:20:50	814	49565
C7	12:20:55	509	44939
C7	12:21:00	513	48084
C7	12:21:05	539	46119
C7	12:21:10	625	46270
Promedio C7		600,00	46995,40
D7	12:21:15	699	39097
D7	12:21:20	714	49470
D7	12:21:25	719	46931
D7	12:21:30	722	46385
D7	12:21:35	723	40790
Promedio D7		715,40	44534,60
E7	12:21:40	722	42166
E7	12:21:45	710	47569
E7	12:21:50	673	40196
E7	12:21:55	646	40222
E7	12:22:00	640	47393
Promedio E7		678,20	43509,20
F7	12:22:05	629	39320
F7	12:22:10	620	40226
F7	12:22:15	608	41830
F7	12:22:20	605	45208
F7	12:22:25	550	47594
Promedio F7		602,40	42835,60
G7	12:22:30	672	40043
G7	12:22:35	680	45562
G7	12:22:40	686	41906
G7	12:22:45	690	40039
G7	12:22:50	686	40169
Promedio G7		682,80	41543,80
H7	12:22:55	690	40006
H7	12:23:00	691	43481
H7	12:23:05	691	47973
H7	12:23:10	693	40746



H7	12:23:15	693	49765
Promedio H7		691,60	44394,20
A8	12:23:20	693	40637
A8	12:23:25	693	41809
A8	12:23:30	690	41670
A8	12:23:35	685	49099
A8	12:23:40	670	48962
Promedio A8		686,20	44435,40
B8	12:23:45	659	43238
B8	12:23:50	590	45374
B8	12:23:55	586	44500
B8	12:24:00	590	38905
B8	12:24:05	591	48791
Promedio B8		603,20	44161,60
C8	12:24:10	591	45171
C8	12:24:15	593	47764
C8	12:24:20	464	38590
C8	12:24:25	550	40707
C8	12:24:30	478	40755
Promedio C8		535,20	42597,40
D8	12:24:35	475	48375
D8	12:24:40	468	44397
D8	12:24:45	364	41518
D8	12:24:50	350	48831
D8	12:24:55	378	47458
Promedio D8		407,00	46115,80
E8	12:25:00	375	40134
E8	12:25:05	368	46474
E8	12:25:10	287	41127
E8	12:25:15	287	42463
E8	12:25:20	287	45016
Promedio E8		320,80	43042,80
F8	12:25:25	287	45210
F8	12:25:30	287	43096
F8	12:25:35	285	41611
F8	12:25:40	274	46561
F8	12:25:45	277	46151
Promedio F8		282,00	44525,80
G8	12:25:50	270	38398
G8	12:25:55	267	43564
G8	12:26:00	273	45503
G8	12:26:05	268	42999
G8	12:26:10	280	38925
Promedio G8		271,60	41877,80
H8	12:26:15	286	46619
H8	12:26:20	285	45123
H8	12:26:25	287	41468
H8	12:26:30	287	39959
H8	12:26:35	287	49789
Promedio H8		286,40	44591,60
A9	12:26:40	287	48749
A9	12:26:45	287	46209
A9	12:26:50	285	43713
A9	12:26:55	274	48696
A9	12:27:00	277	41443
Promedio A9		282,00	45762,00
B9	12:27:05	270	38796



B9	12:27:10	267	41895
B9	12:27:15	273	47730
B9	12:27:20	268	41775
B9	12:27:25	280	47034
Promedio B9		271,60	43446,00
C9	12:27:30	286	49820
C9	12:27:35	285	39085
C9	12:27:40	284	38245
C9	12:27:45	285	45626
C9	12:27:50	285	46099
Promedio C9		285,00	43775,00
D9	12:27:55	286	47451
D9	12:28:00	287	42846
D9	12:28:05	287	48750
D9	12:28:10	287	42610
D9	12:28:15	287	40278
Promedio D9		286,80	44387,00
E9	12:28:20	287	47622
E9	12:28:25	285	42814
E9	12:28:30	274	41358
E9	12:28:35	277	43639
E9	12:28:40	270	49445
Promedio E9		278,60	44975,60
F9	12:28:45	267	40495
F9	12:28:50	273	46982
F9	12:28:55	268	47230
F9	12:29:00	280	41854
F9	12:29:05	286	47828
Promedio F9		274,80	44877,80
G9	12:29:10	285	38625
G9	12:29:15	284	39298
G9	12:29:20	285	45375
G9	12:29:25	285	47082
G9	12:29:30	286	44597
Promedio G9		285,00	42995,40
H9	12:29:35	284	49667
H9	12:29:40	284	38659
H9	12:29:45	285	42735
H9	12:29:50	285	40176
H9	12:29:55	286	40822
Promedio H9		284,80	42411,80
PROMEDIO:		1528.33	43538,033

Fuente: El autor

CONCLUSIÓN

Mediante las mediciones de los niveles de iluminación con el sensor BH1750, se determinó que el aula A-3-2-2 tiene un nivel de iluminación media de 1528.33 lux, cuando la iluminación exterior en un ambiente despejado es de 43538.033 lux.

FOTOGRAFÍAS



Figura 3. Mediciones del aula en donde se va a realizar la implementación del prototipo

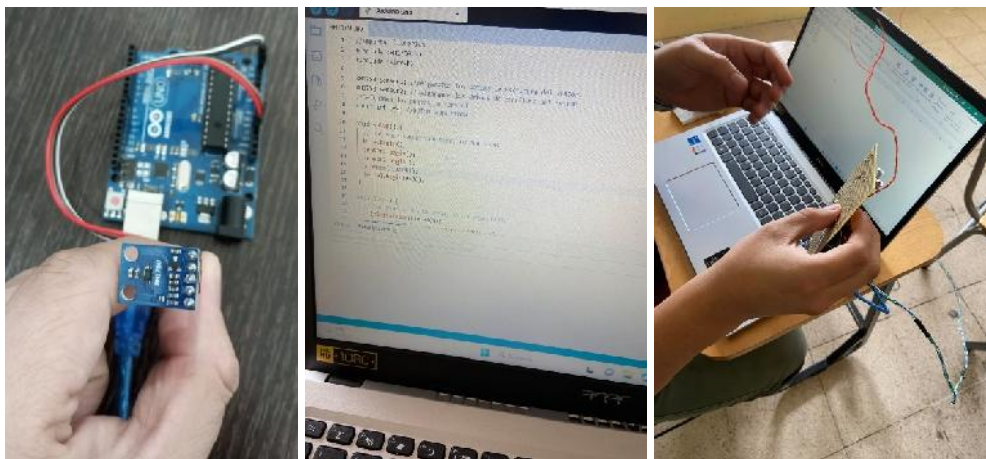


Figura 4. Equipo utilizado para realizar las mediciones de intensidad luminosa



Figura 5. Trazo y marcación de la malla para realizar las mediciones



Figura 6. Primeras mediciones realizadas con el equipo



Figura 7. Segundas mediciones realizadas con el equipo



unl

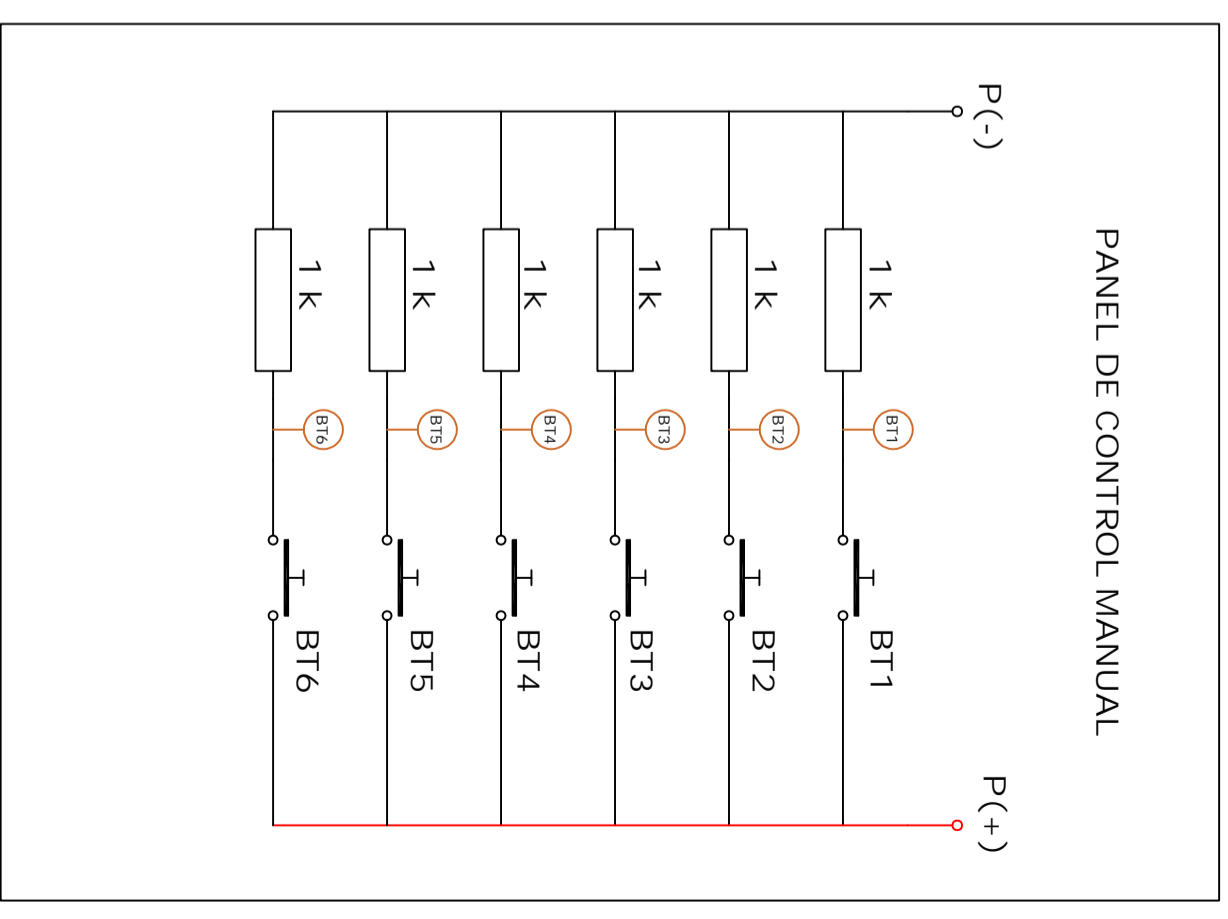
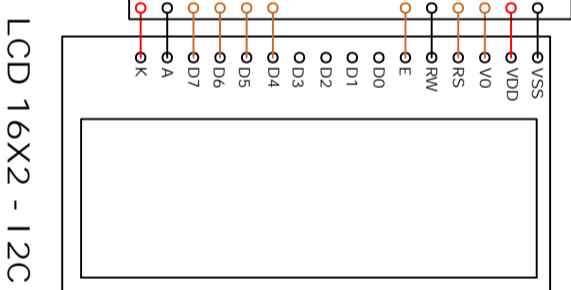
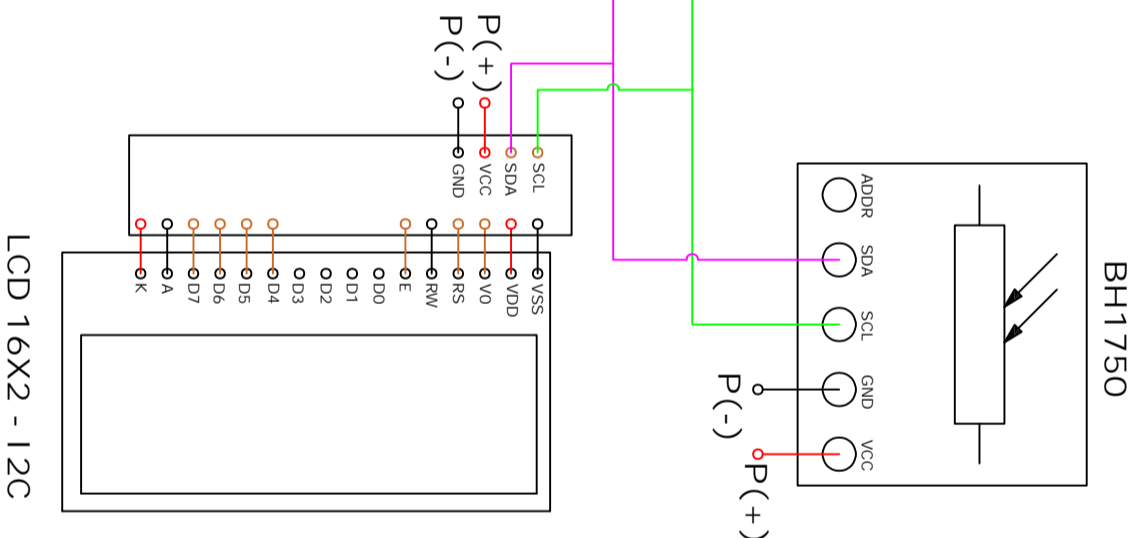
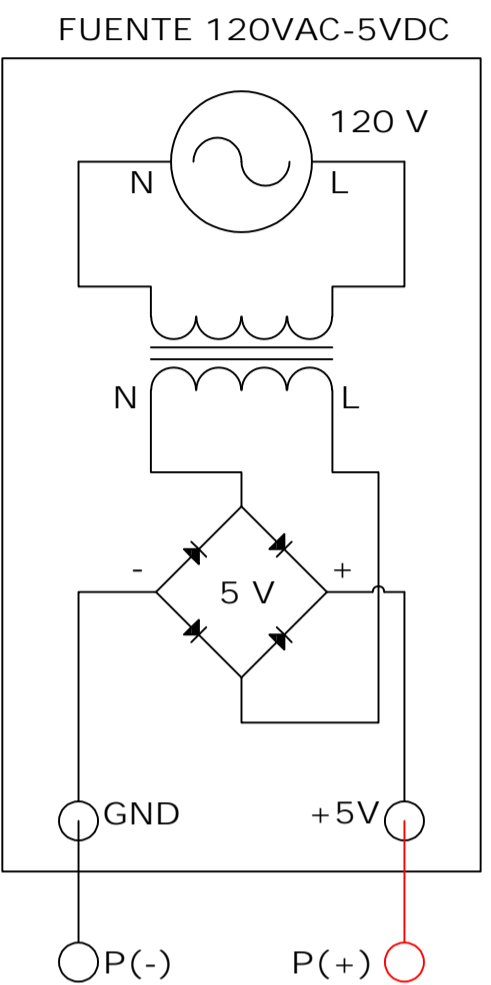
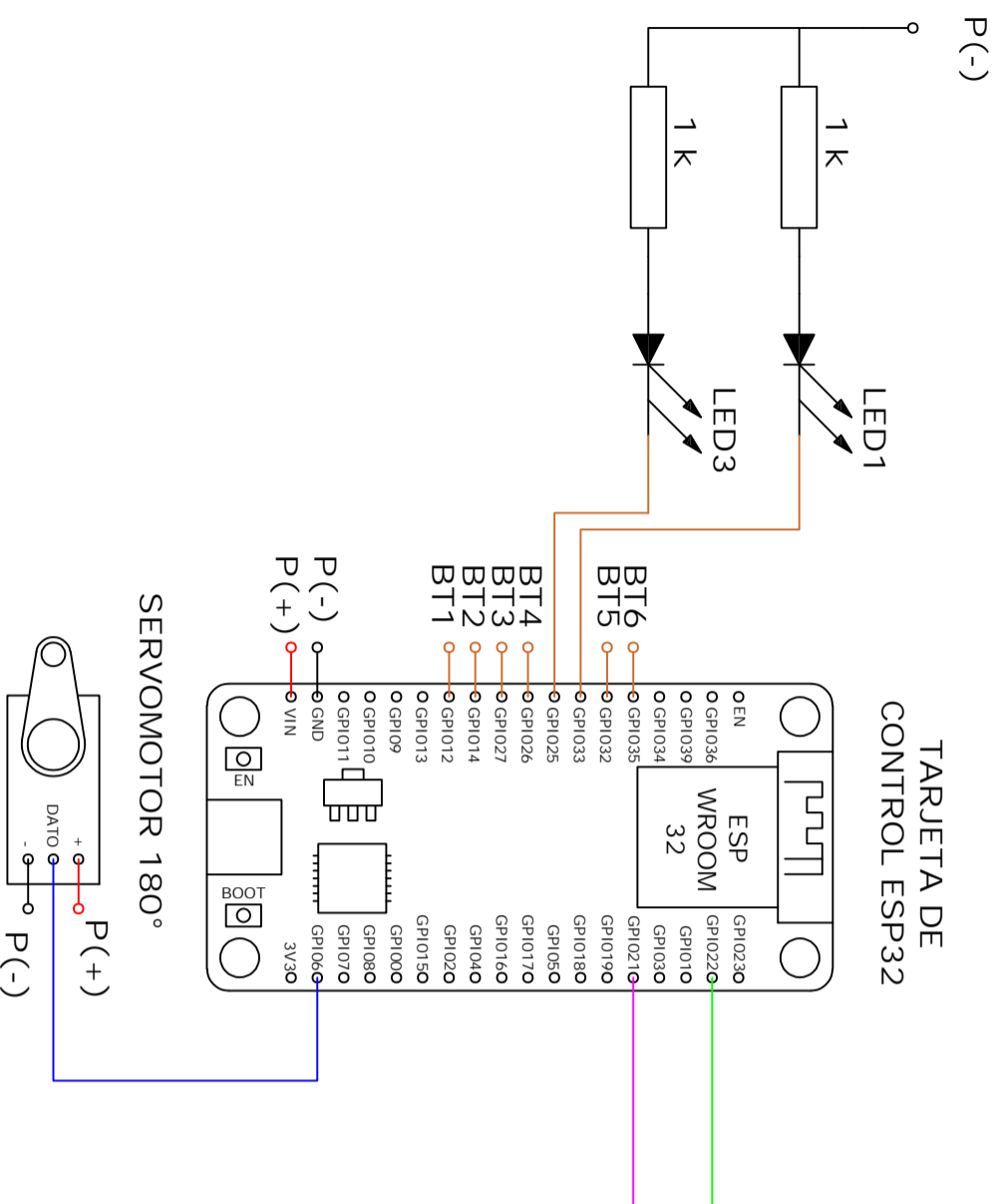
Universidad
Nacional
de Loja

Diseño y construcción de un prototipo de persiana con ajuste automático de iluminación para las aulas de bloque 3 de la FEIRNNR



Figura 8. Terceras mediciones realizadas con el equipo

Anexo 2: Planos de investigación

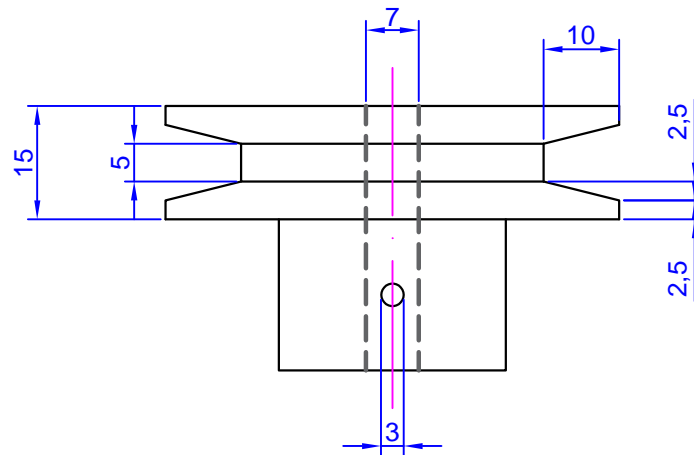
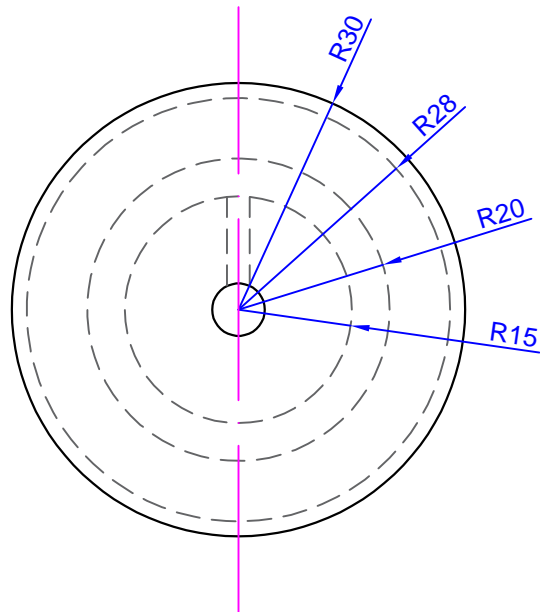


No. de pieza	Descripción del material	N.º de norma o dibujo	Cantidad	Numero del modelo	Observaciones
8	LCD 16X2 - 12C		1		
7	Fuente de 120VAC/5VDC		1		
6	Led color amarillo	IEC 60617	2		
5	Luxómetro BH1750	IEC 60617	1		
4	Resistencia 1 k	IEC 60617	8		
2	ESP32 WROOM		1		
1	Botones azules	IEC 60617	2		

Tolerancia	Peso	Denominación	Escala
		DIAGRAMA DE CONEXIÓN	
		Numero de dibujo	
		01.01.03	

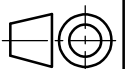
Edi- ción	Modifica- ción	Fecha	Nombre
			Jimmy Carreño
			Fernando Ra.
			Marco Rojas

Cilindrado, C1.6



				Tolerancia	Peso 593g	Materiales Grilon Ø60 mm	
					Fecha	Nombre	Denominación POLEA DEL NEMA17
				Dib.		Jimmy Carreño	
				Rev.		Fernando Ra.	
				Aprob.		Marco Rojas	
							Número de dibujo 01.01.02
Edición	Modificación	Fecha	Nombre				

Escala
1:1



Anexo 3: Algoritmo de programación en Arduino

```

//importando librerías
#include <ESP32Servo.h>
#include <analogWrite.h>
#include <ESP32Tone.h>
#include <ESP32PWM.h>
#include <Wire.h>
#include <BH1750.h> //libreria se sensor
#include <WiFi.h> //incluida librería de internet
#include <FirebaseESP32.h> //librería de Fairebase

//Datos necesarios para conectarse a internet
#define ssid "NETLIFE-CENPOL" //nombre de la rede de internet
#define pass "Cenpol2023*" //contraseña de la red

//Conectar con la base de datos de Firebase
#define URL "periot-1aed6-default-rtdb.firebaseio.com" //Pegar la url de la base de
datos
#define secreto "75kb526uqi7riPlfXTfRhEaJHmQhJJwrMi3eKA31" //pegar el secreto de la
Firebase,
//para ello se debe generar el token Json.
FirebaseData myFireBaseData; //Llevamos los datos a la extensión Firebase
FirebaseJson myJson; //Utilizamos la línea de comunicación Json
FirebaseJsonData myJsonData; //Vinculamos la comunicación con la base de datos

//Para el caso del led necesitamos leer señales digitales
String estados, estado1, estado2, estado3, estado4; //Declaramos las variables iniciales
int intervalo=10000; //Intervalo de tiempo de lectura del sensor
unsigned long actual=0, ultimo=0; //Establecer los valores de lectura de los rangos
float lux; //Asignamos una variable flotante a la iluminación

Servo servoMotor; //asiganamos los datos de escritura del servomotor
BH1750 sensor; //asiganamos los datos de escritura del sensor

//Definimos los puntos de control
const int Bt1=12; //Manual
const int Bt2=14; //Automático
const int Bt3=27; //Abrir
const int Bt4=26; //Serrar
const int LED1=25; //Indicador manual
const int LED2=33; //Indicador automático
int x=0;
bool z=false;
bool y=false;
int lmb=500; //iluminación base de la calibración, la iluminación interior llega a 500

void setup() {
  // Valores de inicio que quiero para arrancar el código
  Serial.begin(115200); //Inicamos el puerto serial pra ver la conexión a la red de la
tarjeta
  Serial.print("Conectando a la red"); //Mensaje mostrado cuando la tarjeta se conecte a
internet
  Serial.print(ssid); //Mostrar mensaje de a que red se conecta
  WiFi.begin(ssid,pass); //Iniciamos el wifi con los datos anteriores

```

```

while(WiFi.status()!=WL_CONNECTED){ //Iniciamos el estado de conexión a la red
  Serial.print("."); //lo que se debe mostrar cuando aun no se conecta a internet
  delay(500); //tiempo en que hace la lectura de la red
}
Firebase.begin(URL,secreto); //Iniciamos la conexión a la base de datos
Firebase.reconnectWiFi(true); //Indicamos que se reconecte a la base de datosn
contantemente en
// caso de desconexión
Serial.print("Conectado con éxito"); //Imprimimos el estado de conexión a la base de
datos
Wire.begin(); //Necesario para el funcionamiento del servo
sensor.begin(); //iniciamos las medidas del sensor
servoMotor.attach(4); //punto de conexión sel servomotor
pinMode(Bt1,INPUT); //declaramos la entrada
pinMode(Bt2,INPUT); //declaramos la entrada
pinMode(Bt3,INPUT); //declaramos la entrada
pinMode(Bt4,INPUT); //declaramos la entrada
pinMode(LED1,OUTPUT); //declaramos al salida
pinMode(LED2,OUTPUT); //declaramos la salida
servoMotor.write(x); //declaramos la variable de control del servo.
}

void loop() {
  //lectura del sensor
  unsigned lux=sensor.readLightLevel(); ///leer lo del sensor BH1750
  Serial.print("Nivel: "); //Imprimir datos en el puerto serial
  Serial.print(lux);
  Serial.println(" lx");
  delay(500);
  Firebase.set(myFireBaseData,"/LECTURAS/ilu",lux); //cargar el dato de iluminación a la
base de datos

  //salida 1
  Firebase.get(myFireBaseData, "/BOTONMANUAL"); //Nos vamos a la base de datos de
Firebase y leemos
  // lo que está en el BOTONLED1
  estados=myFireBaseData.jsonString(); // asignar a estado la comunicación obtenida de
Firebase y Json
  myJson.setJsonData(estados); // Interpretar con la comunicación Json lo leído en la
variable estado
  myJson.get(myJsonData,"/valor"); //El valor que se necetita de la varaible valor de
ese espacio
  // de almacenamiento de la Firebase
  estado1=myJsonData.stringValue; //asiganos ese dato al estado1 (Manual)

  //salida 2 ---- se repite el proceso anterior
  Firebase.get(myFireBaseData, "/BOTONAUTO");
  estados=myFireBaseData.jsonString();
  myJson.setJsonData(estados);
  myJson.get(myJsonData,"/valor");
  estado2=myJsonData.stringValue; //guradamos ese dato al estado2 (Automático)

  //salida 3 ---- se repite el proceso anterior

```

```

Firebase.get(myFireBaseData, "/BOTONABRIR");
estados=myFireBaseData.jsonString();
myJson.setJsonData(estados);
myJson.get(myJsonData,"/valor");
estado3=myJsonData.stringValue; //guradamos ese dato al estado3 (Abrir)

//salida 4 ---- se repite el proceso anterior
Firebase.get(myFireBaseData, "/BOTONCERRAR");
estados=myFireBaseData.jsonString();
myJson.setJsonData(estados);
myJson.get(myJsonData,"/valor");
estado4=myJsonData.stringValue; //guradamos ese dato al estado4 (Cerrar)

//Manual
if(digitalRead(12)==HIGH){ //leer si el boton de manual esta activado
  z=true; //manual
  y=false; //automatico
  Serial.println("modo manual");
  Firebase.setBool(myFireBaseData,"/BOTONAUTO/valor",false); //en firebase automática
desactivado
  delay(500);
}
if(z==true||estado1=="true"){ //leer si el botnon manual en el control o en la app
están activados
  digitalWrite(LED1,HIGH); //se prende el led de manual
  digitalWrite(LED2,LOW); //se apaga el led de automático
  if(digitalRead(27)==HIGH||estado3=="true"){ //la persiana se cierra
    Firebase.setBool(myFireBaseData,"/BOTONCERRAR/valor",false); //en firebase abrir
desactivado
    x=x+5; //incremento de 5 en cada ángulo
    servoMotor.write(x);
    delay(200);
    Serial.println(x);
    if(x>=180){
      x=180; //para el incremento de giro hata los 180 grado (verificar si se va a
trabajar con servo)
    }
  }
}
if(digitalRead(26)==HIGH||estado4=="true"){ //la persiana se abre
  Firebase.setBool(myFireBaseData,"/BOTONABRIR/valor",false); //en firebase cerrar
desactivado
  x=x-5; //reducción de 5 en cada ángulo
  servoMotor.write(x);
  delay(200);
  Serial.println(x);
  if(x<=0){
    x=0; //deternerse hasta que llegue a cero
  }
}
}

//Automático
if(digitalRead(14)==HIGH){ //leer el boton de manual en la tarjeta

```

```

z=false;
y=true;
Serial.println("Modo automático");
Firebase.setBool(myFireBaseData,"/BOTONMANUAL/valor",false);
delay(500);
}
if(y==true||estado2=="true"){ //en el caso que el boton del control o del app estan
activados
digitalWrite(LED2,HIGH); //encender indicador manula
digitalWrite(LED1,LOW); //apagar el indicador de automático
if(lux<250){ //la persiaba se cierra completamente y se queda cerrada
x=180;
Serial.println("iluminación exterior insuficiente");
servoMotor.write(x);
delay(1000);
Serial.println(x);
}
if(lux>250 && lux<1mb){ //la persiana se va ajustando de acuerdo a la escala
x=0; //la persiana se abre completamente
Serial.println("ajustando a 500 lux");
servoMotor.write(x);
delay(1000);
Serial.println(x);
}
if(lux>1mb){
x=0.21667*lux+199.1667; //calibrar con datos reales
Serial.println("cerrando automáticamente"); //la persiana se va cerrando hasta los
500 lux
servoMotor.write(x);
delay(1000);
Serial.println(x);
}
}
}
}
}

```

Anexo 4: Algoritmo de programación en Java: Android Studio

```

package com.example.periot;

import android.os.Bundle;

import com.airbnb.lottie.LottieAnimationView;
import com.google.android.material.snackbar.Snackbar;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.view.View;

import androidx.core.view.WindowCompat;
import androidx.navigation.NavController;
import androidx.navigation.Navigation;
import androidx.navigation.ui.AppBarConfiguration;
import androidx.navigation.ui.NavigationUI;

import com.example.periot.databinding.ActivityMainBinding;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import android.view.Menu;
import android.view.MenuItem;
import android.widget.Button;
import android.widget.TextView;

import java.util.HashMap;

public class MainActivity extends AppCompatActivity {

    //Importar los datos

    TextView lux; //vincular el id para mostrar los luxes
    Button manual, automatico, abrir, cerrar; //vincular los id de los botones
    LottieAnimationView animationView; //vincular la animación

    boolean estado1=false, estado2=false, estado3=false, estado4=false,
estado5=false;
    //las variables de estado de operación

    FirebaseDatabase database=FirebaseDatabase.getInstance();
    DatabaseReference referenceEntrada1=database.getReference("BOTONMANUAL");
    DatabaseReference referenceEntrada2=database.getReference("BOTONAUTO");
    DatabaseReference referenceEntrada3=database.getReference("BOTONABRIR");
    DatabaseReference referenceEntrada4=database.getReference("BOTONCERRAR");
    DatabaseReference referenceEntrada5=database.getReference("LECTURAS");

    private AppBarConfiguration appBarConfiguration;
    private ActivityMainBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        lux=findViewById(R.id.lux);
        manual=findViewById(R.id.manual);
        automatico=findViewById(R.id.automatico);
        abrir=findViewById(R.id.abrir);
        cerrar=findViewById(R.id.cerrar);
        animationView=findViewById(R.id.animationView);
    }
}

```

```

//lectura boton Manual
referenceEntrada1.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        String estadoLed1=snapshot.child("valor").getValue().toString();
        if(estadoLed1.equals("true")){

manual.setBackground(getResources().getDrawable(R.drawable.botonpulsado,null));
        }else{

cerrar.setBackground(getResources().getDrawable(R.drawable.botonnopulsado,null));
        }
        }

    @Override
    public void onCancelled(@NonNull DatabaseError error) {

    }

});

//lectura boton automatico
referenceEntrada2.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        String estadoLed1=snapshot.child("valor").getValue().toString();
        if(estadoLed1.equals("true")){

automatico.setBackground(getResources().getDrawable(R.drawable.botonpulsado,null));
        }else{

cerrar.setBackground(getResources().getDrawable(R.drawable.botonnopulsado,null));
        }
        }

    @Override
    public void onCancelled(@NonNull DatabaseError error) {

    }

});

//lectura del boton abrir
referenceEntrada3.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        String estadoLed1=snapshot.child("valor").getValue().toString();
        if(estadoLed1.equals("true")){

abrir.setBackground(getResources().getDrawable(R.drawable.botonpulsado,null));
        }else{

cerrar.setBackground(getResources().getDrawable(R.drawable.botonnopulsado,null));
        }
        }

    @Override
    public void onCancelled(@NonNull DatabaseError error) {

    }

});

//lectura del boton cerrar
referenceEntrada4.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        String estadoLed1=snapshot.child("valor").getValue().toString();
        if(estadoLed1.equals("true")){

```



```

cerrar.setBackground(getResources().getDrawable(R.drawable.botonpulsado,null));
        }else{

cerrar.setBackground(getResources().getDrawable(R.drawable.botonnopulsado,null));
        }

        @Override
        public void onCancelled(@NonNull DatabaseError error) {

    }

});

//Escribir datos en base de datos
manual.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        HashMap<Object,Object> info=new HashMap<>();
        estado1=!estado1;
        info.put("valor",estado1);
        referenceEntrada1.setValue(info);
        HashMap<Object,Object> infom=new HashMap<>();
        estado2=estado5;
        infom.put("valor",estado2);
        referenceEntrada2.setValue(infom);
    }
});

//Escribir datos en base de datos
automatico.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        HashMap<Object,Object> info=new HashMap<>();
        estado1=estado5;
        info.put("valor",estado1);
        referenceEntrada1.setValue(info);
        HashMap<Object,Object> infom=new HashMap<>();
        estado2=!estado2;
        infom.put("valor",estado2);
        referenceEntrada2.setValue(infom);
    }
});

//Escribir datos en base de datos
abrir.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        HashMap<Object,Object> info=new HashMap<>();
        estado3=!estado3;
        info.put("valor",estado3);
        referenceEntrada3.setValue(info);
        HashMap<Object,Object> infom=new HashMap<>();
        estado4=estado5;
        infom.put("valor",estado4);
        referenceEntrada4.setValue(infom);
    }
});

//Escribir datops en base de datos
cerrar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        HashMap<Object,Object> info=new HashMap<>();
        estado3=estado5;
        info.put("valor",estado3);
        referenceEntrada3.setValue(info);
        HashMap<Object,Object> infom=new HashMap<>();
    }
});

```

```
        estado4=!estado4;
        infom.put("valor",estado4);
        referenceEntrada4.setValue(infom);
    });
}

//Lectura de los valores del sensor
referenceEntrada5.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        String L=snapshot.child("ilu").getValue().toString();
        lux.setText(L);
        animationView.playAnimation();
    }

    @Override
    public void onCancelled(@NonNull DatabaseError error) {
    }
});
}
}
```

Anexo 5: Datasheet del sensor BH1750

Ambient Light Sensor IC Series

Digital 16bit Serial Output Type Ambient Light Sensor IC


BH1750FVI

No.11046EDT01

●Descriptions

BH1750FVI is an digital Ambient Light Sensor IC for I²C bus interface. This IC is the most suitable to obtain the ambient light data for adjusting LCD and Keypad backlight power of Mobile phone. It is possible to detect wide range at High resolution. (1 - 65535 lx).

●Features

- 1) I²C bus Interface (f / s Mode Support)
- 2) Spectral responsibility is approximately human eye response
- 3) Illuminance to Digital Converter
- 4) Wide range and High resolution. (1 - 65535 lx)
- 5) Low Current by power down function
- 6) 50Hz / 60Hz Light noise reject-function
- 7) 1.8V Logic input interface
- 8) No need any external parts
- 9) Light source dependency is little. (ex. Incandescent Lamp. Fluorescent Lamp. Halogen Lamp. White LED. Sun Light)
- 10) It is possible to select 2 type of I²C slave-address.
- 11) Adjustable measurement result for influence of optical window
(It is possible to detect min. 0.11 lx, max. 100000 lx by using this function.)
- 12) Small measurement variation (+/- 20%)
- 13) The influence of infrared is very small.

●Applications

Mobile phone, LCD TV, NOTE PC, Portable game machine, Digital camera, Digital video camera, PDA, LCD display

●Absolute Maximum Ratings

Parameter	Symbol	Ratings	Units
Supply Voltage	Vmax	4.5	V
Operating Temperature	Topr	-40~85	°C
Storage Temperature	Tstg	-40~100	°C
SDA Sink Current	I _{max}	7	mA
Power Dissipation	Pd	260*	mW

* 70mm × 70mm × 1.6mm glass epoxy board. Derating is done at 3.47mW/°C for operating above Ta=25°C.

●Operating Conditions

Parameter	Symbol	Ratings			Units
		Min.	Typ.	Max.	
Vcc Voltage	Vcc	2.4	3.0	3.6	V
I ² C Reference Voltage	V _{DVI}	1.65	-	Vcc	V

●Electrical Characteristics (Vcc = 3.0V, DVI = 3.0V, Ta = 25°C, unless otherwise noted)

Parameter	Symbol	Limits			Units	Conditions
		Min.	Typ.	Max.		
Supply Current	Icc1	—	120	190	μA	Ev = 100 lx ※ ¹
Powerdown Current	Icc2	—	0.01	1.0	μA	No input Light
Peak Wave Length	λp	—	560	—	nm	
Measurement Accuracy	S/A	0.96	1.2	1.44	times	Sensor out / Actual lx EV = 1000 lx ※ ¹ , ※ ²
Dark (0 lx) Sensor out	S0	0	0	3	count	H-Resolution Mode ※ ³
H-Resolution Mode Resolution	rHR	—	1	—	lx	
L-Resolution Mode Resolution	rLR	—	4	—	lx	
H-Resolution Mode Measurement Time	tHR	—	120	180	ms	
L-Resolution Mode Measurement Time	tLR	—	16	24	ms	
Incandescent / Fluorescent Sensor out ratio	rIF	—	1	—	times	EV = 1000 lx
ADDR Input 'H' Voltage	VAH	0.7 * VCC	—	—	V	
ADDR Input 'L' Voltage	VAL	—	—	0.3 * VCC	V	
DVI Input 'L' Voltage	VDVL	—	—	0.4	V	
SCL, SDA Input 'H' Voltage 1	VIH1	0.7 * DVI	—	—	V	DVI ≥ 1.8V
SCL, SDA Input 'H' Voltage 2	VIH2	1.26	—	—	V	1.65V ≤ DVI < 1.8V
SCL, SDA Input 'L' Voltage 1	VIL1	—	—	0.3 * DVI	V	DVI ≥ 1.8V
SCL, SDA Input 'L' Voltage 2	VIL2	—	—	DVI - 1.26	V	1.65V ≤ DVI < 1.8V
SCL, SDA, ADDR Input 'H' Current	IiH	—	—	10	μA	
SCL, SDA, ADDR Input 'L' Current	IiL	—	—	10	μA	
I ² C SCL Clock Frequency	fSCL	—	—	400	kHz	
I ² C Bus Free Time	tBUF	1.3	—	—	μs	
I ² C Hold Time (repeated) START Condition	tHDSTA	0.6	—	—	μs	
I ² C Set up time for a Repeated START Condition	tsUSTA	0.6	—	—	μs	
I ² C Set up time for a Repeated STOP Condition	tsUSTD	0.6	—	—	μs	
I ² C Data Hold Time	tHDDAT	0	—	0.9	μs	
I ² C Data Setup Time	tsUDAT	100	—	—	ns	
I ² C 'L' Period of the SCL Clock	tLOW	1.3	—	—	μs	
I ² C 'H' Period of the SCL Clock	tHIGH	0.6	—	—	μs	
I ² C SDA Output 'L' Voltage	VOL	0	—	0.4	V	IOL = 3 mA

※¹ White LED is used as optical source.

※² Measurement Accuracy typical value is possible to change '1' by "Measurement result adjustment function".

※³ Use H-resolution mode or H-resolution mode2 if dark data (less than 10 lx) is need.

●Reference Data

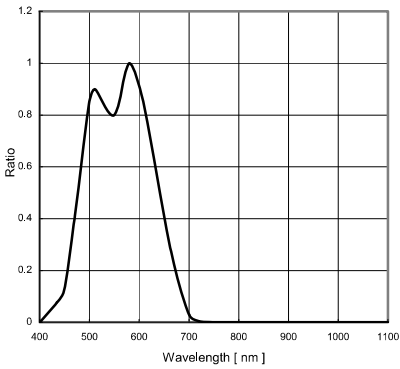


Fig.1 Spectral Response

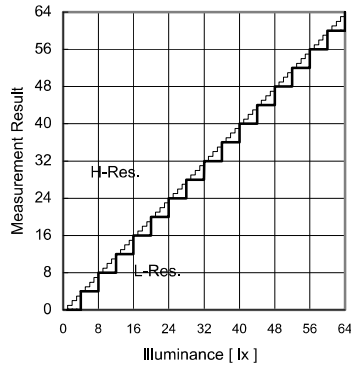


Fig.2 Illuminance - Measurement Result 1

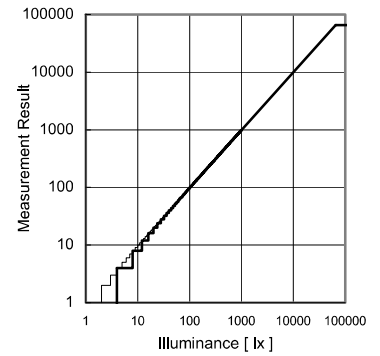


Fig.3 Illuminance - Measurement Result 2

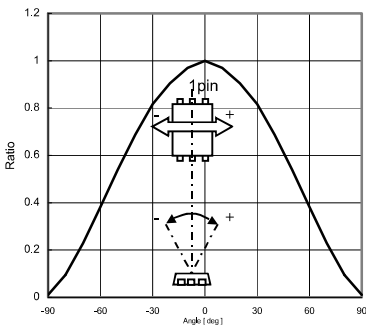


Fig.4 Directional Characteristics 1

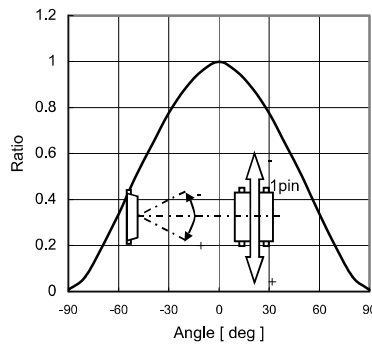


Fig.5 Directional Characteristics 2

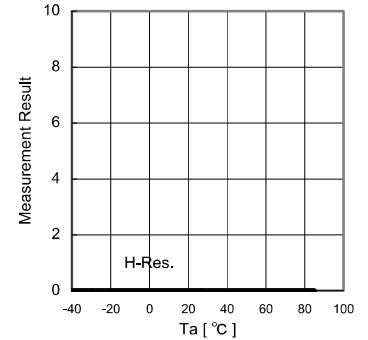


Fig.6 Dark Response

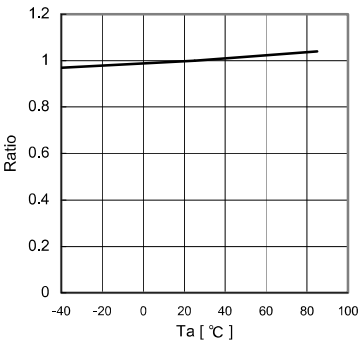


Fig.7 Measurement Accuracy Temperature Dependency

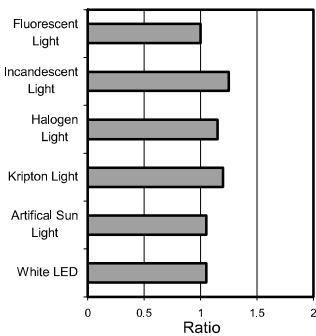


Fig.8 Light Source Dependency (Fluorescent Light is set to '1')

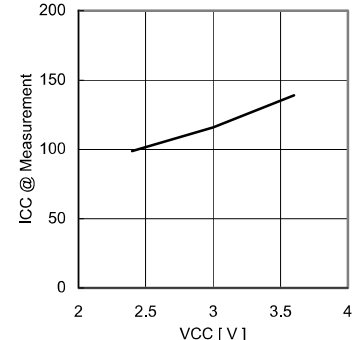


Fig.9 VCC - ICC (During measurement)

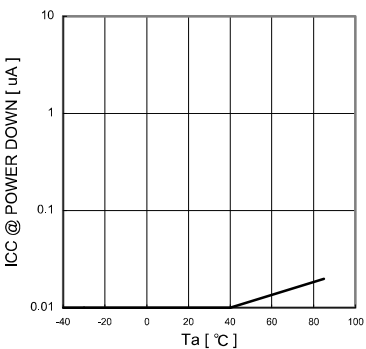


Fig.10 VCC - ICC@0 Lx (POWER DOWN)

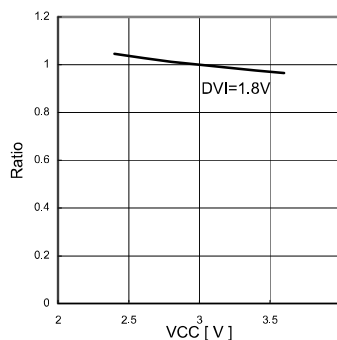


Fig.11 Measurement Result VCC Dependency

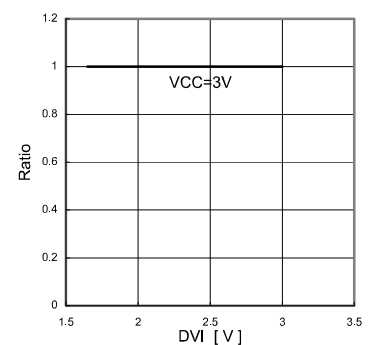
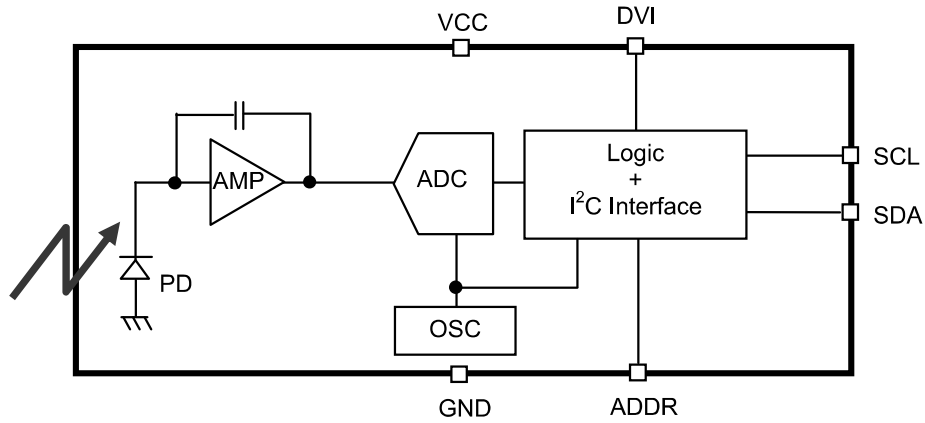


Fig.12 Measurement Result DVI Dependency

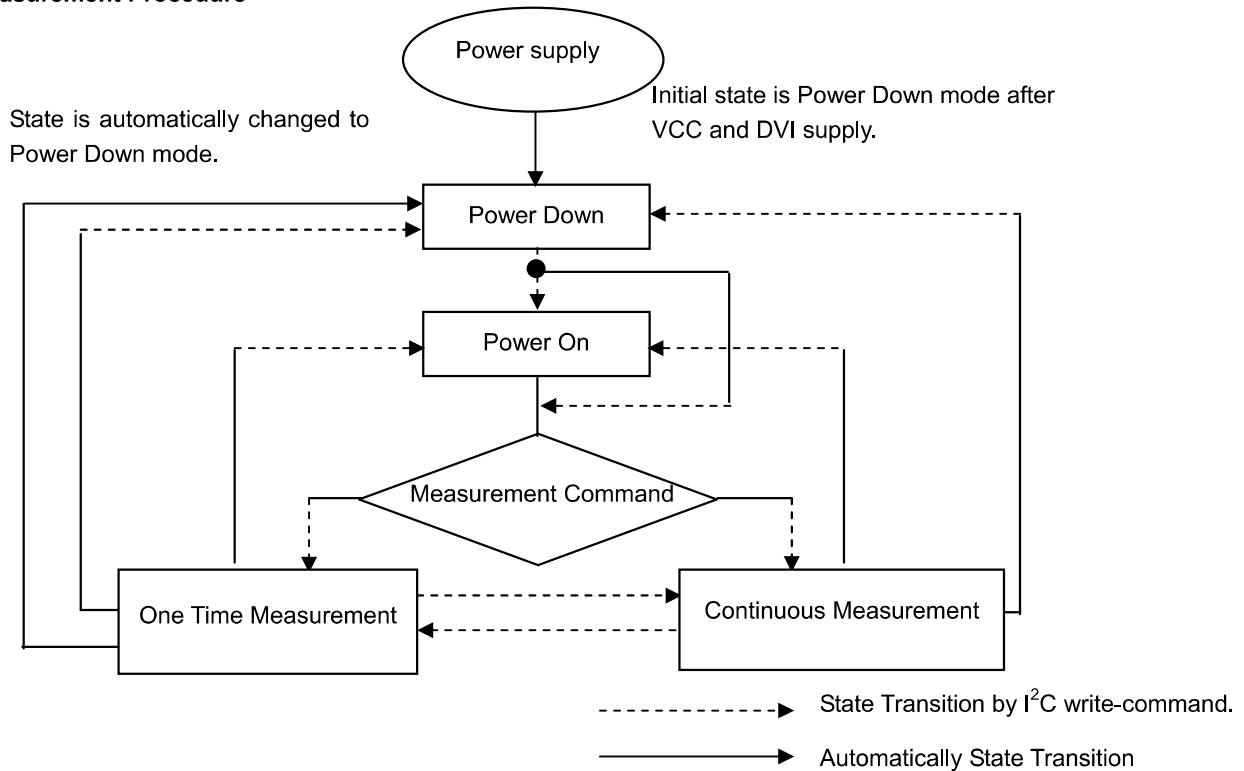
●Block Diagram



●Block Diagram Descriptions

- PD
Photo diode with approximately human eye response.
- AMP
Integration-OPAMP for converting from PD current to Voltage.
- ADC
AD converter for obtainment Digital 16bit data.
- Logic + I²C Interface
Ambient Light Calculation and I²C BUS Interface. It is including below register.
Data Register → This is for registration of Ambient Light Data. Initial Value is "0000_0000_0000_0000".
Measurement Time Register → This is for registration of measurement time. Initial Value is "0100_0101".
- OSC
Internal Oscillator (typ. 320kHz). It is CLK for internal logic.

●Measurement Procedure



* "Power On" Command is possible to omit.

●Instruction Set Architecture

Instruction	Opecode	Comments
Power Down	0000_0000	No active state.
Power On	0000_0001	Waiting for measurement command.
Reset	0000_0111	Reset Data register value. Reset command is not acceptable in Power Down mode.
Continuously H-Resolution Mode	0001_0000	Start measurement at 1lx resolution. Measurement Time is typically 120ms.
Continuously H-Resolution Mode2	0001_0001	Start measurement at 0.5lx resolution. Measurement Time is typically 120ms.
Continuously L-Resolution Mode	0001_0011	Start measurement at 4lx resolution. Measurement Time is typically 16ms.
One Time H-Resolution Mode	0010_0000	Start measurement at 1lx resolution. Measurement Time is typically 120ms. It is automatically set to Power Down mode after measurement.
One Time H-Resolution Mode2	0010_0001	Start measurement at 0.5lx resolution. Measurement Time is typically 120ms. It is automatically set to Power Down mode after measurement.
One Time L-Resolution Mode	0010_0011	Start measurement at 4lx resolution. Measurement Time is typically 16ms. It is automatically set to Power Down mode after measurement.
Change Measurement time (High bit)	01000_MT[7,6,5]	Change measurement time. ※ Please refer "adjust measurement result for influence of optical window."
Change Masurement time (Low bit)	011_MT[4,3,2,1,0]	Change measurement time. ※ Please refer "adjust measurement result for influence of optical window."

※ Don't input the other opecode.

●Measurement mode explanation

Measurement Mode	Measurement Time.	Resolurtion
H-resolution Mode2	Typ. 120ms.	0.5 lx
H-Resolution Mode	Typ. 120ms.	1 lx.
L-Resolution Mode	Typ. 16ms.	4 lx.

We recommend to use H-Resolution Mode.

Measurement time (integration time) of H-Resolution Mode is so long that some kind of noise(including in 50Hz / 60Hz noise) is rejected. And H-Resolution Mode is 1 lx resolution so that it is suitable for darkness (less than 10 lx)
H-resolution mode2 is also suitable to detect for darkness.

●Explanation of Asynchronous reset and Reset command "0000_0111"

1) Asynchronous reset

All registers are reset. It is necessary on power supply sequence. Please refer "Timing chart for VCC and DVI power supply sequence" in this page. It is power down mode during DVI = 'L'.

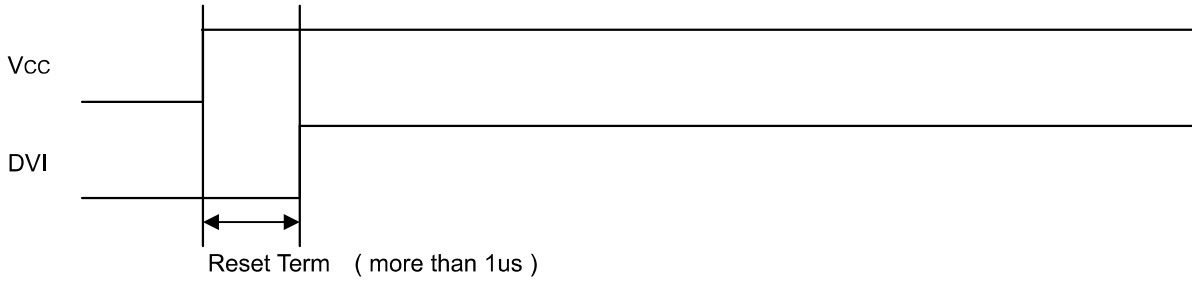
2) Reset command

Reset command is for only reset Illuminance data register. (reset value is '0') It is not necessary even power supply sequence.It is used for removing previous measurement result. This command is not working in power down mode, so that please set the power on mode before input this command.

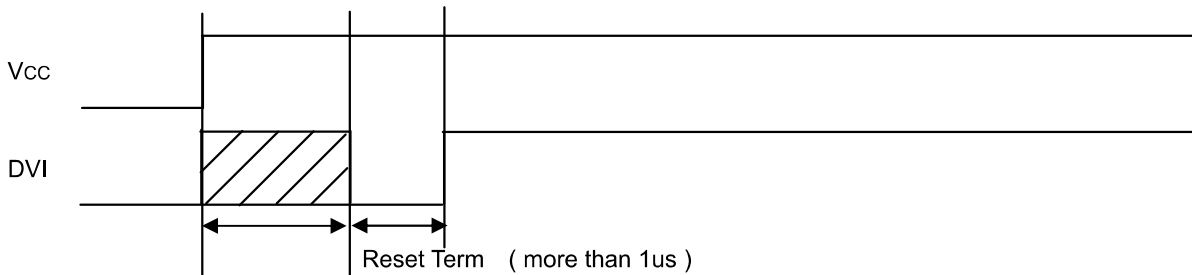
●Timing chart for VCC and DVI power supply sequence

DVI is I²C bus reference voltage terminal. And it is also asynchronous reset terminal. It is necessary to set to 'L' after Vcc is supplied. In DVI 'L' term, internal state is set to Power Down mode.

1) Recommended Timing chart1 for VCC and DVI supply.



2) Timing chart2 for VCC and DVI supply.
(If DVI rises within 1μs after VCC supply)



Don't care state

ADDR, SDA, SCL is not stable if DVI 'L' term (1us) is not given by systems.

In this case, please connect the resistors (approximately 100kOhm) to ADDR without directly connecting to VCC or GND, because it is 3 state buffer for Internal testing.

●Measurement sequence example from "Write instruction" to "Read measurement result"

ex1) Continuously H-resolution mode (ADDR = 'L')



from Master to Slave



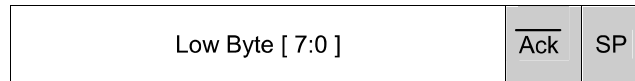
from Slave to Master

① Send "Continuously H-resolution mode " instruction



② Wait to complete 1st H-resolution mode measurement.(max. 180ms.)

③ Read measurement result.



How to calculate when the data High Byte is "10000011" and Low Byte is "10010000"
 $(2^{15} + 2^9 + 2^8 + 2^7 + 2^4) / 1.2 \doteq 28067 [lx]$

The result of continuously measurement mode is updated.(120ms.typ at H-resolution mode, 16ms.typ at L-resolution mode)

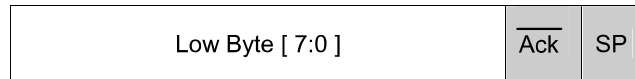
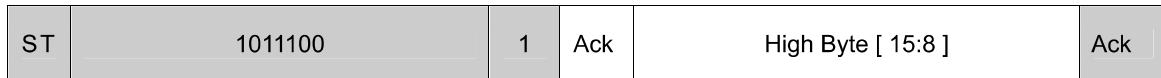
ex2) One time L-resolution mode (ADDR = 'H')

① Send "One time L-resolution mode " instruction



② Wait to complete L-resolution mode measurement.(max. 24ms.)

③ Read measurement result



How to calculate when the data High Byte is "00000001" and Low Byte is "00010000"
 $(2^8 + 2^4) / 1.2 \doteq 227 [lx]$

In one time measurement, Statement moves to power down mode after measurement completion.If updated result is need then please resend measurement instruction.

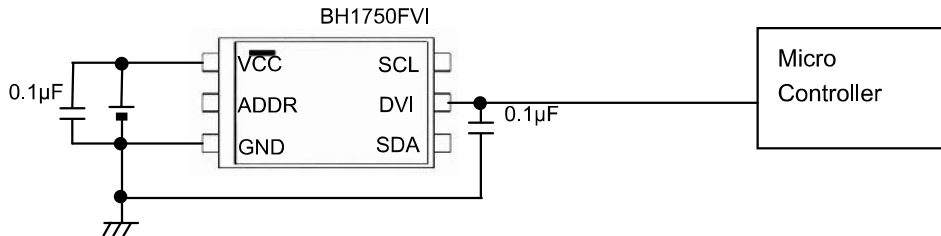
●Application circuit example of DVI terminal

The DVI terminal is an asynchronous reset terminal. Please note that there is a possibility that IC doesn't operate normally if the reset section is not installed after the start-up of Vcc.

(Please refer to the paragraph of "Timing chart for Vcc and DVI power supply sequence")

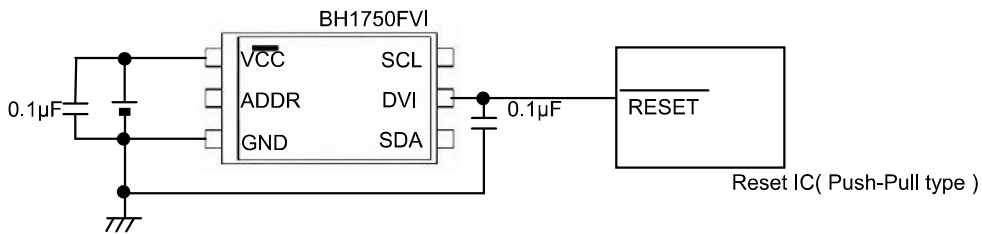
The description concerning SDA and the terminal SCL is omitted in this application circuit example. Please design the application the standard of the I2C bus as it finishes being satisfactory. Moreover, the description concerning the terminal ADDR is omitted. Please refer to the paragraph of "Timing chart for Vcc and DVI power supply sequence" about the terminal ADDR design.

ex 1) The control signal line such as CPU is connected.

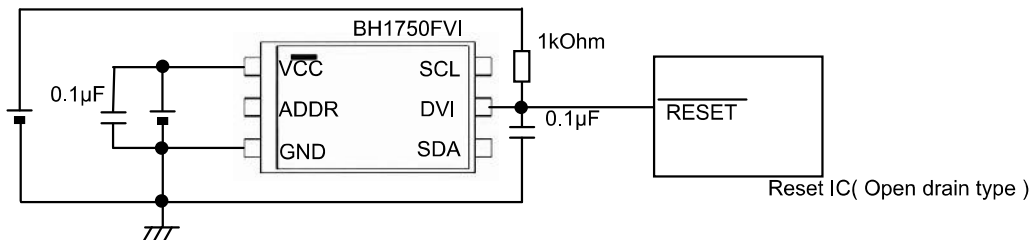


ex 2) Reset IC is used.

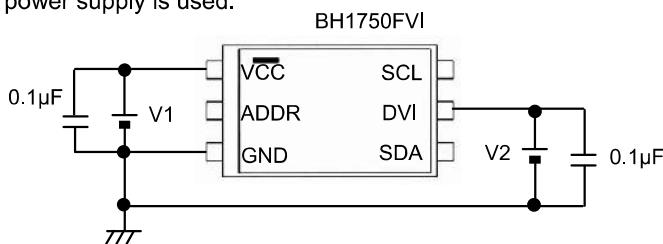
1, For Reset IC of the Push-Pull type



2, For Reset IC of the Open drain output



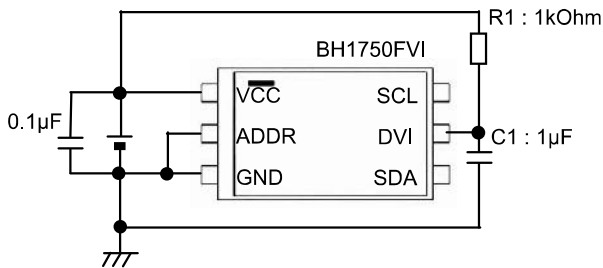
ex 3) A different power supply is used.



※ Power supply of DVI must stand up later than power supply of VCC stand up, because it is necessary to secure reset section (1µs or more).

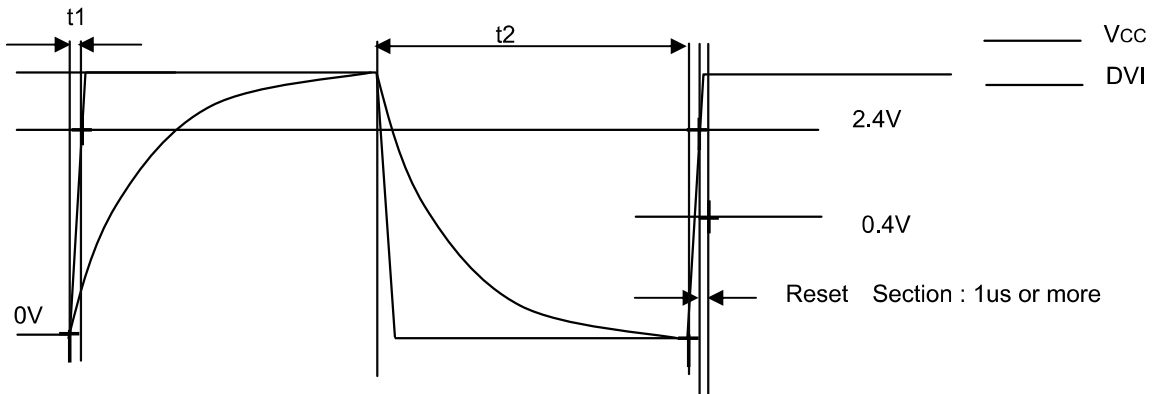
ex 4) LPF using CR is inserted between VCC and DVI.

This method has the possibility that the Reset section of turning on the power supply can not satisfied. cannot be satisfied. Please design the set considering the characteristic of the power supply enough.



◆ Notes when CR is inserted between VCC and DVI

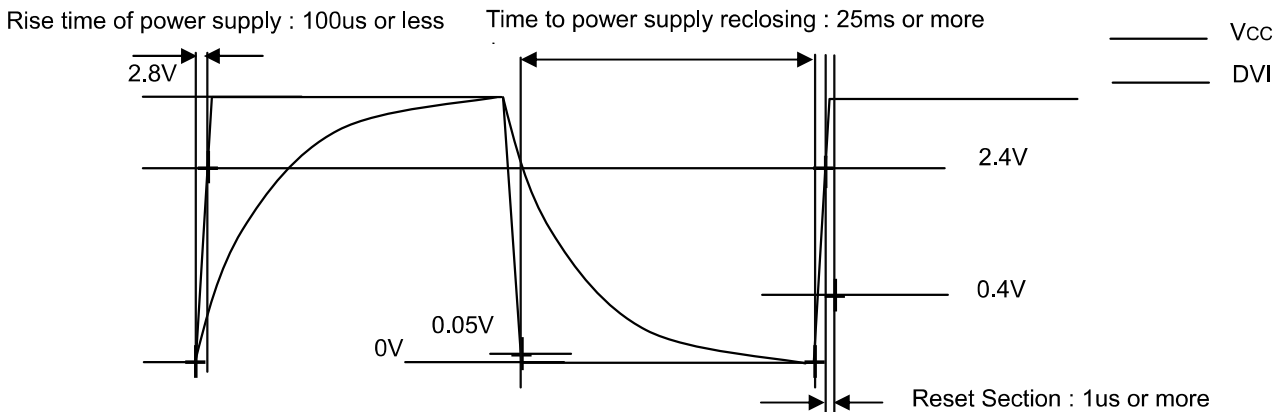
- ※ Please note that there is a possibility that reset section (1µs) can not be satisfied because the power supply is turned on when the rise time of VCC is slow
- ※ When VCC is turned off, the DVI voltage becomes higher than VCC voltage but IC destruction is not occred if recommended constant (R1 = 1kOhm, C1 = 1µF) is used.
- ※ Please note that there is a possibility that Reset section (1µsec) cannot be satisfied if wait time is not enough long after turning off VCC. (It is necessary to consider DVI voltage level after turning off VCC.)



• Please do the application design to secure Reset section 1us or more after the reclosing of the power supply.

◆ Example of designing set when CR (C = 1µF, R = 1kΩ) is inserted between VCC and DVI with VCC=2.8V

- ①The rise time to 0→2.4V of VCC must use the power supply of 100µs or less.
- ②Please wait 25ms or more after VCC turn off (VCC <= 0.05V), because it is necessary to secure reset section (1µs or more).

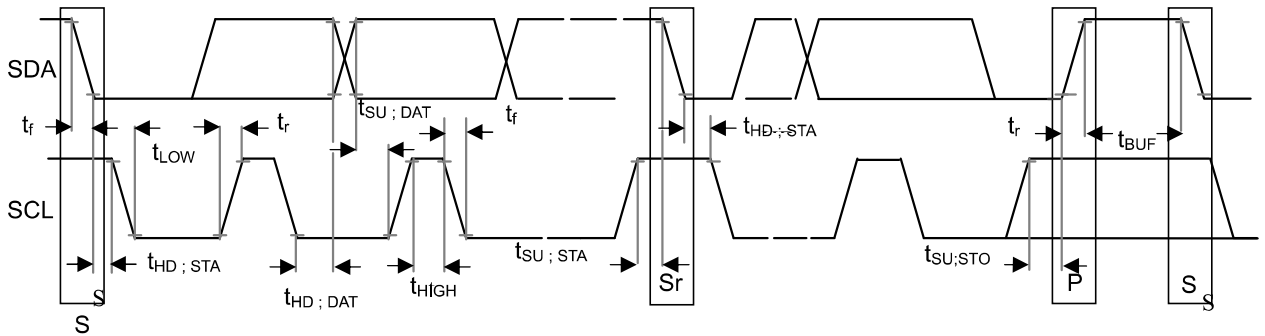


• Please do the application design to secure Reset section 1us or more after the reclosing of the power supply.

● I²C Bus Access

1) I²C Bus Interface Timing chart

Write measurement command and Read measurement result are done by I²C Bus interface. Please refer the formally specification of I²C Bus interface, and follow the formally timing chart.



2) Slave Address

Slave Address is 2 types, it is determined by ADDR Terminal

ADDR = 'H' (ADDR \geq 0.7VCC) \rightarrow "1011100"

ADDR = 'L' (ADDR \leq 0.3VCC) \rightarrow "0100011"

3) Write Format

BH1750FVI is not able to accept plural command without stop condition. Please insert SP every 1 Opcode.

ST	Slave Address	R/W 0	Ack	Opcode	Ack	SP
----	---------------	----------	-----	--------	-----	----

4) Read Format

ST	Slave Address	R/W 1	Ack	High Byte [15:8] $2^{15} \ 2^{14} \ 2^{13} \ 2^{12} \ 2^{11} \ 2^{10} \ 2^9 \ 2^8$	Ack
----	---------------	----------	-----	---	-----

$2^7 \ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0$	Low Byte [7:0]	$\overline{\text{Ack}}$	SP
---	----------------	-------------------------	----



from Master to Slave



from Slave to Master

ex)

High Byte = "1000_0011"

Low Byte = "1001_0000"

$$(2^{15} + 2^9 + 2^8 + 2^7 + 2^4) / 1.2 \doteq 28067 [lx]$$

* I²C BUS is trademark of Phillips Semiconductors. Please refer formality specification.

●Adjust measurement result for influence of optical window. (sensor sensitivity adjusting)

BH1750FVI is possible to change sensor sensitivity. And it is possible to cancel the optical window influence (difference with / without optical window) by using this function. Adjust is done by changing measurement time. For example, when transmission rate of optical window is 50% (measurement result becomes 0.5 times if optical window is set), influence of optical window is ignored by changing sensor sensitivity from default to 2 times

Sensor sensitivity is shift by changing the value of MTreg (measurement time register). MTreg value has to set 2 times if target of sensor sensitivity is 2 times. Measurement time is also set 2 times when MTreg value is changed from default to 2 times.

ex) Procedure for changing target sensor sensitivity to 2 times.

Please change Mtrege from "0100_0101" (default) to "1000_1010" (default * 2).

1) Changing High bit of MTreg

ST	Slave Address	R/W 0	Ack	01000_100	Ack	SP
----	---------------	----------	-----	-----------	-----	----

2) Changing Low bit of MTreg

ST	Slave Address	R/W 0	Ack	011_01010	Ack	SP
----	---------------	----------	-----	-----------	-----	----

3) Input Measurement Command

ST	Slave Address	R/W 0	Ack	0001_0000	Ack	SP
----	---------------	----------	-----	-----------	-----	----

* This example is High Resolution mode, but it accepts the other measurement.

4) After about 240ms, measurement result is registered to Data Register.
(High Resolution mode is typically 120ms, but measurement time is set twice.)

The below table is seeing the changable range of MTreg.

		Min.	Typ.	Max.
changeable range of MTreg	binary	0001_1111 (sensitivity : default * 0.45)	0100_0101 default	1111_1110 (sensitivity : default * 3.68)
	decimal	31 (sensitivity : default * 0.45)	69 default	254 (sensitivity : default * 3.68)

It is possilbe to detect 0.23lx by using this function at H-resolution mode. And it is possilbe to detect 0.11lx by using this function at H-resolution mode2.

The below formula is to calculate illuminance per 1 count.

H-reslution mode : Illuminance per 1 count (lx / count) = 1 / 1.2 * (69 / X)
 H-reslution mode2 : Illuminance per 1 count (lx / count) = 1 / 1.2 * (69 / X) / 2

1.2 : Measurement accuracy
 69 : Default value of MTreg (dec)
 X : MTreg value

The below table is seeing the detail of resolution.

Mtreg の値	lx / count at H-resolutin mode	lx / count at H-resolution mode2
0001_1111	1.85	0.93
0100_0101	0.83	0.42
1111_1110	0.23	0.11

●H-Resolution Mode2

H-resolution mode2 is 0.5lx (typ.) resolution mode. It is suitable if under less than 10 lx measurement data is necessary. This measurement mode supports " Adjust measurement result for influence of optical window ". Please refer it. It is possible to detect min. 0.11 lx by using H-resolution mode2.

○ Instruction set architecture for H-resolution mode2

Instruction	Opecode	Comments
Continuously H-Resolution Mode2	0001_0001	Start measurement at 0.5lx resolution. Measurement Time is typically 120ms.
One Time H-Resolution Mode2	0010_0001	Start measurement at 0.5lx resolution. Measurement Time is typically 120ms. It is automatically set to Power Down mode after measurement.

○ Measurement sequence example from "Write instruction" to "Read measurement result"

ex) Continuously H-resolution mode2 (ADDR = 'L')

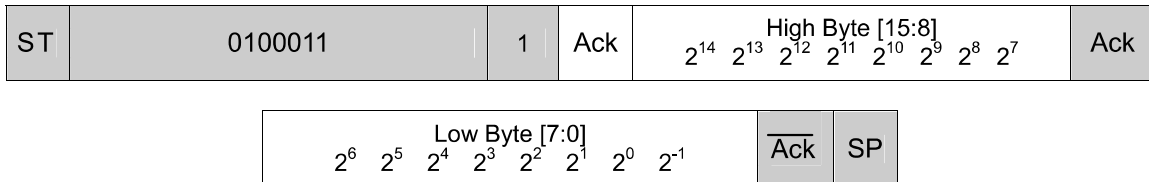


① Send "Continuously H-resolution mode2 " instruction



② Wait to complete 1st H-resolution mode2 measurement.(max. 180ms.)

③ Read measurement result.



How to calculate when the data High Byte is "00000000" and Low Byte is "00010010"

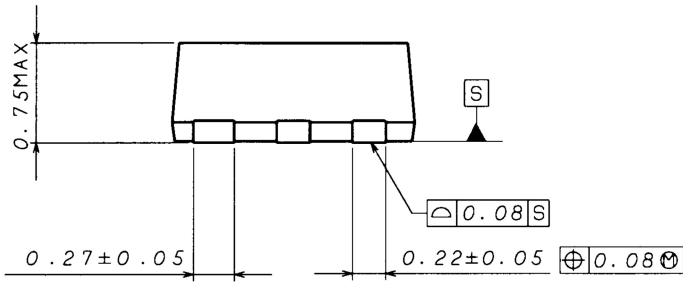
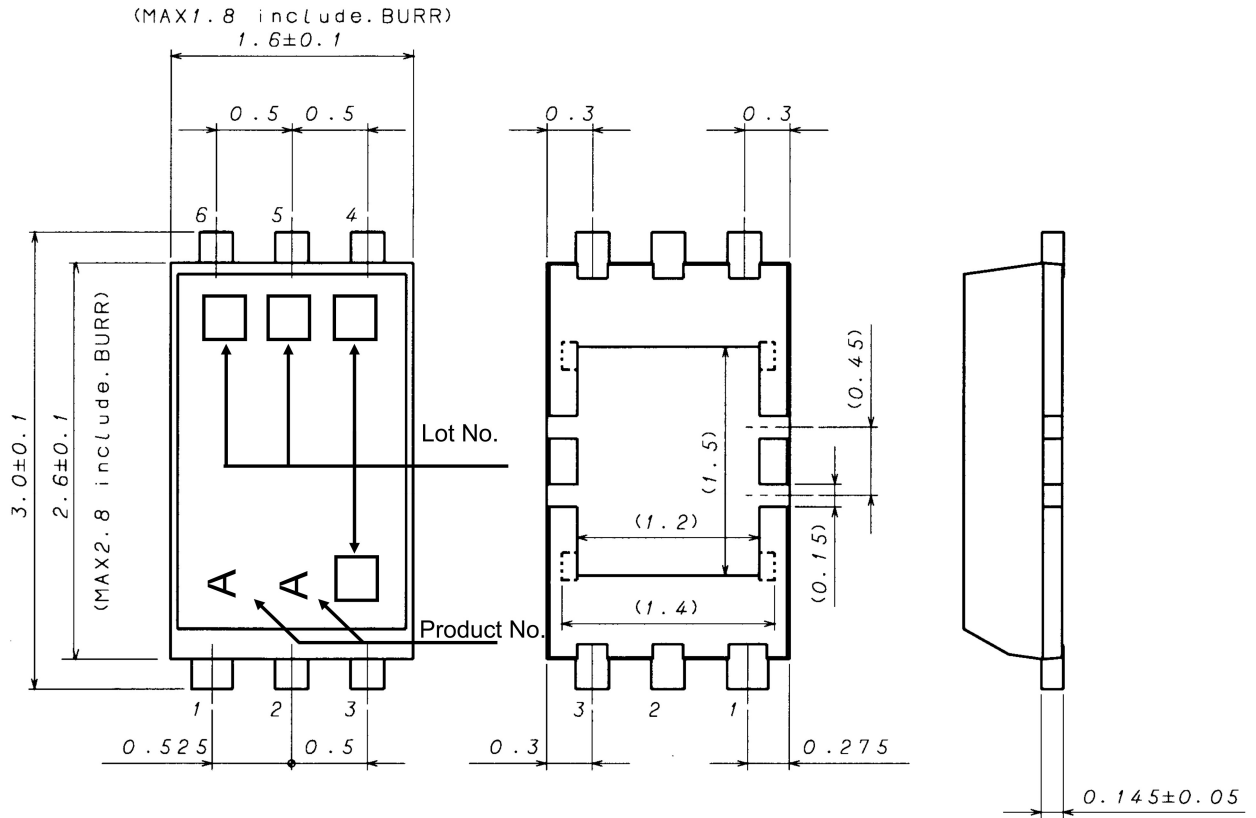
$$(2^3 + 2^0) / 1.2 \doteq 7.5 [lx]$$

●Terminal Description

PIN No.	Terminal Name	Equivalent Circuit	Function
1	VCC		Power Supply Terminal
2	ADDR		<p>I²C Slave-address Terminal ADDR = 'H' (ADDR ≥ 0.7Vcc) "1011100" ADDR = 'L' (ADDR ≤ 0.3Vcc) "0100011" ADDR Terminal is designed as 3 state buffer for internal test. So that please take care of Vcc and DVI supply procedure. Please see P6.</p>
3	GND		GND Terminal
4	SDA		I ² C bus Interface SDA Terminal
5	DVI		<p>SDA, SCL Reference Voltage Terminal And DVI Terminal is also asynchronous Reset for internal registers. So that please set to 'L' (at least 1μs, DVI ≤ 0.4V) after Vcc is supplied. BH1750FVI is pulled down by 150kOhm while DVI = 'L'.</p>
6	SCL		I ² C bus Interface SCL Terminal

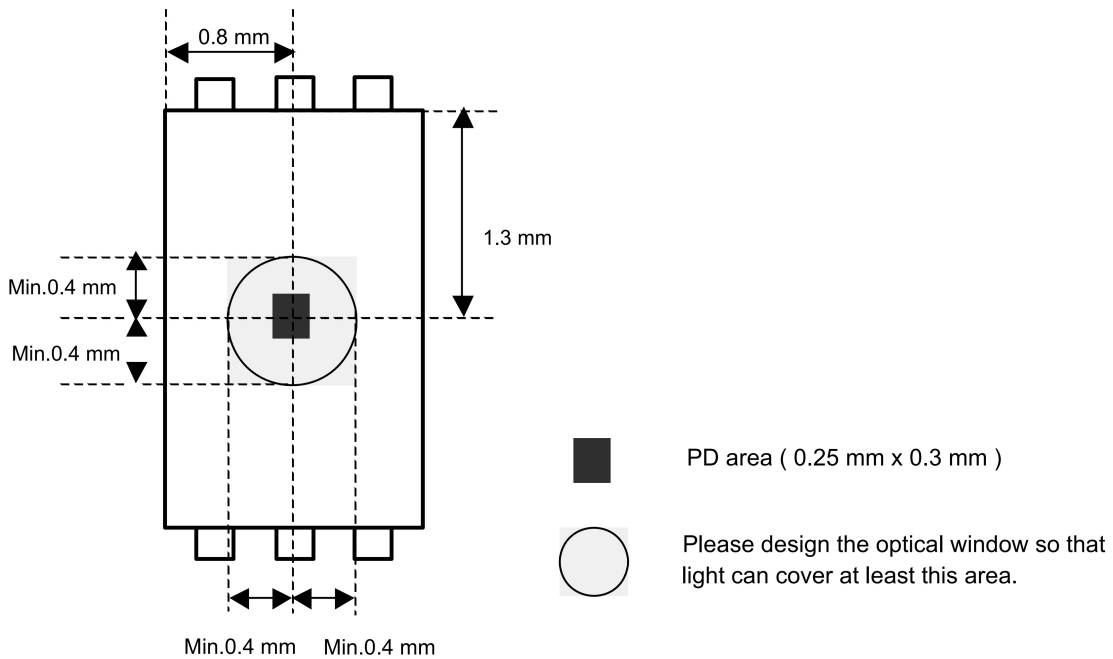
※These values are design-value, not guaranteed.

●Package Outlines



WSOF6I (Unit : mm)

●About an optical design on the device

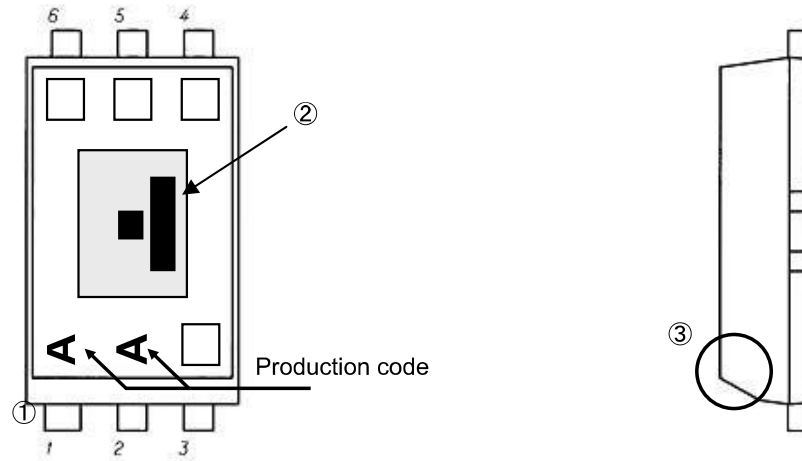


●The method of distinguishing 1pin.

There is some method of distinguishing 1pin.

- ① Distinguishing by 1Pin wide-lead
- ② Distinguishing by die pattern
- ③ Distinguishing by taper part of 1-3pin side

② (by die pattern) is the easiest method to distinguish by naked eye.



●Cautions on use

1) Absolute Maximum Ratings

An excess in the absolute maximum ratings, such as supply voltage (V_{max}), temperature range of operating conditions (T_{opr}), etc., can break down devices, thus making impossible to identify breaking mode such as a short circuit or an open circuit. If any special mode exceeding the absolute maximum ratings is assumed, consideration should be given to take physical safety measures including the use of fuses, etc.

2) GND voltage

Make setting of the potential of the GND terminal so that it will be maintained at the minimum in any operating state. Furthermore, check to be sure no terminals are at a potential lower than the GND voltage including an actual electric transient.

3) Short circuit between terminals and erroneous mounting

In order to mount ICs on a set PCB, pay thorough attention to the direction and offset of the ICs. Erroneous mounting can break down the ICs. Furthermore, if a short circuit occurs due to foreign matters entering between terminals or between the terminal and the power supply or the GND terminal, the ICs can break down.

4) Operation in strong electromagnetic field

Be noted that using ICs in the strong electromagnetic field can malfunction them.

5) Inspection with set PCB

On the inspection with the set PCB, if a capacitor is connected to a low-impedance IC terminal, the IC can suffer stress. Therefore, be sure to discharge from the set PCB by each process. Furthermore, in order to mount or dismount the set PCB to/from the jig for the inspection process, be sure to turn OFF the power supply and then mount the set PCB to the jig. After the completion of the inspection, be sure to turn OFF the power supply and then dismount it from the jig. In addition, for protection against static electricity, establish a ground for the assembly process and pay thorough attention to the transportation and the storage of the set PCB.

6) Input terminals

In terms of the construction of IC, parasitic elements are inevitably formed in relation to potential. The operation of the parasitic element can cause interference with circuit operation, thus resulting in a malfunction and then breakdown of the input terminal. Therefore, pay thorough attention not to handle the input terminals; such as to apply to the input terminals a voltage lower than the GND respectively, so that any parasitic element will operate. Furthermore, do not apply a voltage to the input terminals when no power supply voltage is applied to the IC. In addition, even if the power supply voltage is applied, apply to the input terminals a voltage lower than the power supply voltage or within the guaranteed value of electrical characteristics.

7) Thermal design

Perform thermal design in which there are adequate margins by taking into account the power dissipation (P_d) in actual states of use.

8) Treatment of package

Dusts or scratch on the photo detector may affect the optical characteristics. Please handle it with care.

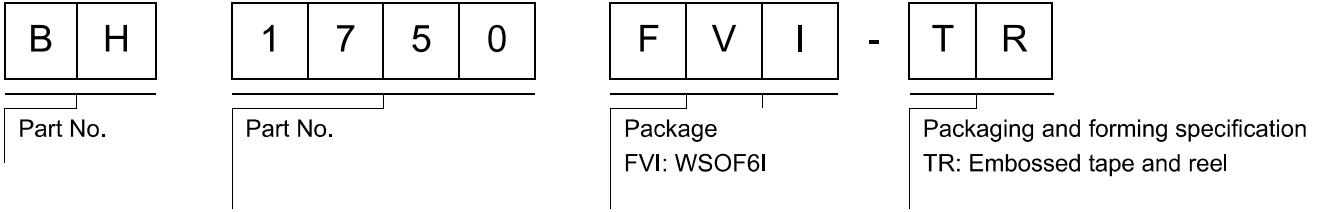
9) Rush current

When power is first supplied to the CMOS IC, it is possible that the internal logic may be unstable and rush current may flow instantaneously. Therefore, give special consideration to power coupling capacitance, power wiring, width of GND wiring, and routing of connections.

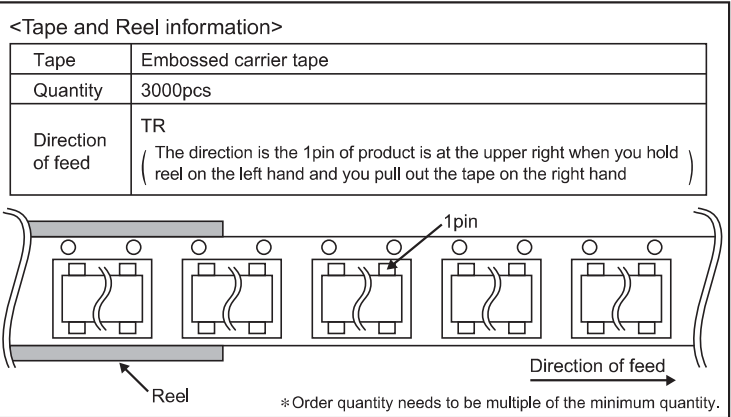
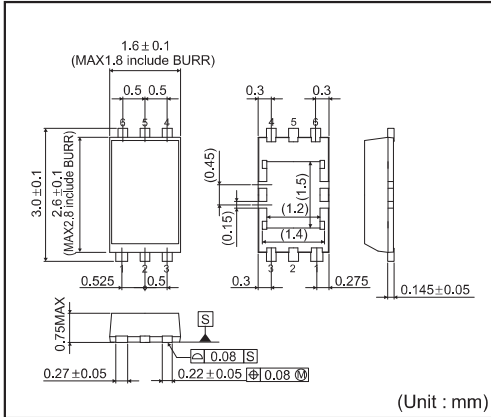
10) The exposed central pad on the back side of the package

There is an exposed central pad on the back side of the package. But please do it non connection. (Don't solder, and don't do electrical connection) Please mount by Footprint dimensions described in the Jisso Information for WSOF6I. This pad is GND level, therefore there is a possibility that LSI malfunctions and heavy-current is generated.

●Ordering part number



WSOF6I



Notice

Precaution on using ROHM Products

- Our Products are designed and manufactured for application in ordinary electronic equipments (such as AV equipment, OA equipment, telecommunication equipment, home electronic appliances, amusement equipment, etc.). If you intend to use our Products in devices requiring extremely high reliability (such as medical equipment ^(Note 1), transport equipment, traffic equipment, aircraft/spacecraft, nuclear power controllers, fuel controllers, car equipment including car accessories, safety devices, etc.) and whose malfunction or failure may cause loss of human life, bodily injury or serious damage to property ("Specific Applications"), please consult with the ROHM sales representative in advance. Unless otherwise agreed in writing by ROHM in advance, ROHM shall not be in any way responsible or liable for any damages, expenses or losses incurred by you or third parties arising from the use of any ROHM's Products for Specific Applications.

(Note1) Medical Equipment Classification of the Specific Applications

JAPAN	USA	EU	CHINA
CLASS III	CLASS III	CLASS II b	CLASS III
CLASS IV		CLASS III	

- ROHM designs and manufactures its Products subject to strict quality control system. However, semiconductor products can fail or malfunction at a certain rate. Please be sure to implement, at your own responsibilities, adequate safety measures including but not limited to fail-safe design against the physical injury, damage to any property, which a failure or malfunction of our Products may cause. The following are examples of safety measures:
 - Installation of protection circuits or other protective devices to improve system safety
 - Installation of redundant circuits to reduce the impact of single or multiple circuit failure
- Our Products are designed and manufactured for use under standard conditions and not under any special or extraordinary environments or conditions, as exemplified below. Accordingly, ROHM shall not be in any way responsible or liable for any damages, expenses or losses arising from the use of any ROHM's Products under any special or extraordinary environments or conditions. If you intend to use our Products under any special or extraordinary environments or conditions (as exemplified below), your independent verification and confirmation of product performance, reliability, etc. prior to use, must be necessary:
 - Use of our Products in any types of liquid, including water, oils, chemicals, and organic solvents
 - Use of our Products outdoors or in places where the Products are exposed to direct sunlight or dust
 - Use of our Products in places where the Products are exposed to sea wind or corrosive gases, including Cl₂, H₂S, NH₃, SO₂, and NO₂
 - Use of our Products in places where the Products are exposed to static electricity or electromagnetic waves
 - Use of our Products in proximity to heat-producing components, plastic cords, or other flammable items
 - Sealing or coating our Products with resin or other coating materials
 - Use of our Products without cleaning residue of flux (even if you use no-clean type fluxes, cleaning residue of flux is recommended); or Washing our Products by using water or water-soluble cleaning agents for cleaning residue after soldering
 - Use of the Products in places subject to dew condensation
- The Products are not subject to radiation-proof design.
- Please verify and confirm characteristics of the final or mounted products in using the Products.
- In particular, if a transient load (a large amount of load applied in a short period of time, such as pulse. is applied, confirmation of performance characteristics after on-board mounting is strongly recommended. Avoid applying power exceeding normal rated power; exceeding the power rating under steady-state loading condition may negatively affect product performance and reliability.
- De-rate Power Dissipation (Pd) depending on Ambient temperature (Ta). When used in sealed area, confirm the actual ambient temperature.
- Confirm that operation temperature is within the specified range described in the product specification.
- ROHM shall not be in any way responsible or liable for failure induced under deviant condition from what is defined in this document.

Precaution for Mounting / Circuit board design

- When a highly active halogenous (chlorine, bromine, etc.) flux is used, the residue of flux may negatively affect product performance and reliability.
- In principle, the reflow soldering method must be used; if flow soldering method is preferred, please consult with the ROHM representative in advance.

For details, please refer to ROHM Mounting specification

Precautions Regarding Application Examples and External Circuits

1. If change is made to the constant of an external circuit, please allow a sufficient margin considering variations of the characteristics of the Products and external components, including transient characteristics, as well as static characteristics.
2. You agree that application notes, reference designs, and associated data and information contained in this document are presented only as guidance for Products use. Therefore, in case you use such information, you are solely responsible for it and you must exercise your own independent verification and judgment in the use of such information contained in this document. ROHM shall not be in any way responsible or liable for any damages, expenses or losses incurred by you or third parties arising from the use of such information.

Precaution for Electrostatic

This Product is electrostatic sensitive product, which may be damaged due to electrostatic discharge. Please take proper caution in your manufacturing process and storage so that voltage exceeding the Products maximum rating will not be applied to Products. Please take special care under dry condition (e.g. Grounding of human body / equipment / solder iron, isolation from charged objects, setting of Ionizer, friction prevention and temperature / humidity control).

Precaution for Storage / Transportation

1. Product performance and soldered connections may deteriorate if the Products are stored in the places where:
 - [a] the Products are exposed to sea winds or corrosive gases, including Cl₂, H₂S, NH₃, SO₂, and NO₂
 - [b] the temperature or humidity exceeds those recommended by ROHM
 - [c] the Products are exposed to direct sunshine or condensation
 - [d] the Products are exposed to high Electrostatic
2. Even under ROHM recommended storage condition, solderability of products out of recommended storage time period may be degraded. It is strongly recommended to confirm solderability before using Products of which storage time is exceeding the recommended storage time period.
3. Store / transport cartons in the correct direction, which is indicated on a carton with a symbol. Otherwise bent leads may occur due to excessive stress applied when dropping of a carton.
4. Use Products within the specified time after opening a humidity barrier bag. Baking is required before using Products of which storage time is exceeding the recommended storage time period.

Precaution for Product Label

QR code printed on ROHM Products label is for ROHM's internal use only.

Precaution for Disposition

When disposing Products please dispose them properly using an authorized industry waste company.

Precaution for Foreign Exchange and Foreign Trade act

Since our Products might fall under controlled goods prescribed by the applicable foreign exchange and foreign trade act, please consult with ROHM representative in case of export.

Precaution Regarding Intellectual Property Rights

1. All information and data including but not limited to application example contained in this document is for reference only. ROHM does not warrant that foregoing information or data will not infringe any intellectual property rights or any other rights of any third party regarding such information or data. ROHM shall not be in any way responsible or liable for infringement of any intellectual property rights or other damages arising from use of such information or data.:
2. No license, expressly or implied, is granted hereby under any intellectual property rights or other rights of ROHM or any third parties with respect to the information contained in this document.

Other Precaution

1. This document may not be reprinted or reproduced, in whole or in part, without prior written consent of ROHM.
2. The Products may not be disassembled, converted, modified, reproduced or otherwise changed without prior written consent of ROHM.
3. In no event shall you use in any way whatsoever the Products and the related technical information contained in the Products or this document for any military purposes, including but not limited to, the development of mass-destruction weapons.
4. The proper names of companies or products described in this document are trademarks or registered trademarks of ROHM, its affiliated companies or third parties.

General Precaution

1. Before you use our Products, you are requested to carefully read this document and fully understand its contents. ROHM shall not be in any way responsible or liable for failure, malfunction or accident arising from the use of any ROHM's Products against warning, caution or note contained in this document.
2. All information contained in this document is current as of the issuing date and subject to change without any prior notice. Before purchasing or using ROHM's Products, please confirm the latest information with a ROHM sales representative.
3. The information contained in this document is provided on an "as is" basis and ROHM does not warrant that all information contained in this document is accurate and/or error-free. ROHM shall not be in any way responsible or liable for any damages, expenses or losses incurred by you or third parties resulting from inaccuracy or errors of or concerning such information.

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

[ROHM Semiconductor:](#)

[BH1750FVI-TR](#)

Anexo 6: Datasheet del 17HS4401

Anexo 7: Certificado de traducción del resumen



FINE-TUNED ENGLISH LANGUAGE INSTITUTE

Líderes en la Enseñanza del Inglés

Ing. María Belén Novillo Sánchez.

ENGLISH TEACHER- FINE TUNED ENGLISH CIA LTDA.

CERTIFICA:

Que el documento aquí compuesto es fiel traducción del idioma español al idioma inglés del resumen del trabajo de integración curricular: **Diseño y construcción de un prototipo de persiana con ajuste automático de iluminación para las aulas del bloque 3 de la FEIRNNR**, autoría del señor **Jimmy Paúl Carreño Ordóñez** con número de cédula **1105228728**, estudiante de la Carrera de Ingeniería Electromecánica de la Universidad Nacional de Loja

Lo certifico en honor a la verdad y autorizo al interesado hacer uso del presente en lo que a sus intereses convenga.

Loja, 15 de noviembre del 2023



Ing. María Belén Novillo Sánchez.

ENGLISH TEACHER- FINE TUNED ENGLISH CIA LTDA.

Matriz - Loja: Macará 205-51 entre Rocafuerte y Miguel Riofrío - Teléfono: 072578899
Zamora: García Moreno y Pasaje 12 de Febrero - Teléfono: 072608169
Yantzaza: Jorge Mosquera y Luis Bastidas - Edificio Sindicato de Choferes - Teléfono: 072301329

www.fte.edu.ec

Anexo 8: Productos disponibles en el mercado

Pasa el mouse encima de la imagen para aplicar zoom

SwitchBot Persianas motorizadas de inclinación ciega, persianas eléctricas inteligentes con control remoto Bluetooth, alimentadas por energía solar, control de detección de luz, añade SwitchBot Hub Mini para que sea compatible con Alexa y Google Home

4.6 ★★★★☆ - 349 calificaciones

US\$69.99

Sin depósito de derechos de importación y US\$29.40 de envío a Ecuador [Detalles](#)

Oferta promocional Ahorra un 30% en Artículo con beneficio ... [Cupones por 3](#)

Compatible con Alexa
Agregue control de voz combinando con un dispositivo y un hub Alexa

[Ver más](#)

Marca	SwitchBot
Color	Bianco
Material	Plástico
Dimensiones del producto	1.92"L x 11.5"W pulgadas

US\$69.00

Sin depósito de derechos de importación y US\$29.40 de envío a Ecuador [Detalles](#)

Entrega el **martes, 22 de agosto**

O entrega más rápida el **jueves, 17 de agosto**. Realiza el pedido en 4 hrs

[Enviar a Ecuador](#)

Disponible

Cantidad: 1

[Agregar al Carrito](#)

[Comprar ahora](#)

Pago: Transacción segura
 Envío por: Amazon
 Vendido por: Wanelo Inc
 Devoluciones: Elegible para devolución, reintegro o sustitución dentro...
 Soporte: Se incluye asistencia gratuita para los

CINONEKO Persianas celulares motorizadas con energía solar, sin taladro, opacas, para ventanas, sin herramientas, persianas inteligentes de panel inalámbricas Alexa y Google, gris, 23 x 64 pulgadas

5.0 ★★★★★ - 2 calificaciones

US\$138.00

Sin depósito de derechos de importación y US\$36.79 de envío a Ecuador [Detalles](#)

Cupón: Aplicar cupón de 50% [Compra artículos](#) | [Términos](#)

US\$138.00

Sin depósito de derechos de importación y US\$36.79 de envío a Ecuador [Detalles](#)

Entrega el **viernes, 25 de agosto**

O entrega más rápida el **jueves, 17 de agosto**. Realiza el pedido en 1 hr

[Enviar a Ecuador](#)

Solo queda(n) 4 en stock (hay

Figura 34: Productos disponibles en el mercado

CORTINASHD
Tienda Online de Cortinas

Buscar...

ACCEDER / REGISTRARSE

CARRITO

CORTINAS ▾ PERSIANAS ▾ CORTINAS MOTORIZADAS ▾ ROLLER IMPRESAS ▾ CORTINAS DE TELA ▾ MECANISMOS ▾ PUERTAS ▾ TOLDOS ▾ MARCAS ▾

Más de 18.000 Cortinas vendidas

Fabricación Rollers 3 a 4 días hábiles

Asesoría gratis Ventas x whatsapp

INICIO / MOTORES PARA CORTINAS

-8% Dcto



Motor para persianas verticales de pvc + control remoto marca Coulisse

~~S/2,590.00~~ **S/2,390.00**

- ✓ Fuente de alimentación: 12-24V DC - 1,5 A
- ✓ Potencia nominal 36 W
- ✓ Fuerza de tracción de la cadena: 26 N.
- ✓ Max. peso persiana vertical: 25 kg / 55 Lb.

AÑADIR AL CARRITO

Figura 35: Controlador domótico para persiana vertical