



Facultad de Ciencia y tecnología

Tecnología Superior en Electrónica Automotriz

Trabajo de Titulación:

Reparación y Calibración de un Luxómetro

Trabajo previo a la obtención del título de Tecnólogo en Electrónica Automotriz

Autores:

César Augusto Batallas López

Bryan Jeyson Pesantez Pesantez

Director:

Ing. Efrén Fernández

Cuenca – Ecuador

2025

Dedicatoria

A ti, Tío Froilán Luna, a quien debo tanto y más. No basta con palabras para expresar lo que siento por todo el apoyo, amor y sabiduría que me has brindado a lo largo de este arduo camino. Tu generosidad, paciencia y sacrificio han sido faros que me han guiado, especialmente en los momentos de incertidumbre y desafío. Cada paso de esta travesía ha estado marcado por tu aliento constante, por tu confianza en mí cuando mis fuerzas flaqueaban y por tu inquebrantable fe en mis sueños.

Este logro no solo es mío, es también tuyo, porque sin tu respaldo incondicional, sin tus palabras sabias y tu ejemplo de perseverancia, esta meta hubiera sido simplemente un sueño lejano. Gracias por estar siempre a mi lado, por ser mi mayor apoyo, y por haber sido una fuente constante de inspiración y motivación.

Te dedico esta tesis con el corazón lleno de gratitud y emoción. Este triunfo, al igual que todos los demás logros que alcanzaremos en el futuro, lleva tu nombre, Tío, porque tú has sido el pilar fundamental de mi éxito.

Att

César Batallas L.

A mis queridos padres, sin ustedes no estaría donde estoy hoy. Esta tesis es el reflejo de su amor incondicional, su sacrificio y su dedicación incansable para darme todo lo que necesitaba para alcanzar mis sueños. Desde mis primeros pasos hasta este logro, siempre han sido mi mayor fuente de inspiración y fortaleza.

Mi gratitud hacia ustedes no tiene límites, pero hoy quiero expresar con todo mi corazón que esta tesis se dedica a ustedes, mis héroes. Les debo todo lo que soy y todo lo que llegaré a ser.

Att

Bryan Pesantez P.

Agradecimientos

Expreso mi más sincero agradecimiento al Ing. Efrén Fernández, cuyo valioso acompañamiento, conocimientos y orientación fueron fundamentales para el desarrollo de este proyecto de titulación. Su experiencia y compromiso con la excelencia académica y técnica han sido un pilar fundamental en la reparación y calibración del luxómetro, asegurando que el proceso se llevara a cabo con rigor metodológico y en conformidad con las normativas internacionales aplicables. Asimismo, deseo agradecer a la Universidad del Azuay, por brindarnos los conocimientos teóricos y las instalaciones. Sin este respaldo institucional, la materialización de este proyecto no habría sido posible.

Finalmente, extendiendo mi gratitud a todas aquellas personas que, de manera directa o indirecta, contribuyeron con su conocimiento, tiempo y apoyo para la culminación de este estudio. Este trabajo no solo representa un logro académico, sino también una contribución a la seguridad vial y a la mejora en los procesos de calibración de sistemas de iluminación vehicular.

Resumen

El presente trabajo constituye un informe técnico sobre la reparación y calibración de un luxómetro. utilizado para la verificación de la intensidad luminosa en la calibración de luces de vehículos. El luxómetro en cuestión presentaba fallas que impedían su correcto funcionamiento, lo que motivó la necesidad de intervenir para restaurar sus capacidades operativas. La metodología aplicada incluyó un diagnóstico detallado de las fallas, la sustitución de componentes defectuosos y la implementación de pruebas para asegurar su correcto desempeño. Posteriormente, se realizarán procesos de calibración basados en normativas internacionales para garantizar que las mediciones de iluminación sean precisas y confiables. La culminación de este trabajo no solo permitió recuperar la funcionalidad del instrumento, sino también validar su conformidad con los estándares requeridos para su uso en la calibración de luces vehiculares, contribuyendo así a la seguridad vial y a la correcta regulación del sistema de iluminación automotriz.

Palabras clave: reparación, calibración, mantenimiento, luxómetro, fotometría.

Abstract:

This project aims to repair and calibrate a lux meter used to verify light intensity in vehicle headlight calibration. The lux meter in question exhibited malfunctions that prevented its proper operation, prompting the need for intervention to restore its functional capabilities. The applied methodology included a detailed diagnosis of the faults, replacement of defective components, and the implementation of tests to ensure correct performance. Subsequently, calibration processes were carried out based on international standards to guarantee accurate and reliable illuminance measurements. The completion of this work not only restored the instrument's functionality but also validated its compliance with the required standards for use in vehicle headlight calibration, thus contributing to road safety and the proper regulation of automotive lighting systems.

Keywords: repair, calibration, maintenance, luxometer, photometry.

Índice De Contenido

Contenido

Dedicatoria	i
Agradecimientos	ii
Resumen.....	iii
Abstract:	iii
Índice De Contenido	iv
Índice de tablas	vi
Índice de figuras	vii
1. Introducción	1
2. Objetivos	2
2.1 Objetivo General.	2
2.2 Objetivos Específicos.....	2
3. Procedimiento	3
3.2. Desarme e Identificación de Componentes	5
3.2.1. Retiro de tornillos y tapas:	5
3.2.2. Desconexión de cables y componentes mecánicos:	5
3.2.3. Extracción de la PCB:	6
3.3. Limpieza General.....	6
3.3.1. Limpieza de la carcasa vacía:.....	6
3.3.2. Limpieza de componentes:	7
3.3.3. Limpieza de la PCB.....	7
Remoción de polvo:.....	7
3.4. Análisis de Componentes Electrónicos de la PCB	8
3.4.1. Problemas identificados durante la limpieza:	8
3.4.2. Revisión de los componentes de la PCB.....	9
3.5. Componentes e Identificación que integran el Luxómetro.	10
3.6. Montaje del luxómetro	12
3.6.1. Montaje de la PCB reparada:	12
3.6.2. Montaje de los componentes Mecánicos:	12
3.6.3. Mejoras realizadas	13
3.6.4. Conexión de componentes:.....	14
3.6.5. Ajuste del potenciómetro:	14
3.6.6. Colocación de tapa:	14
3.7. Prueba de Encendido y Medición.....	14

3.8. Preparación de placa electrónica nueva	15
3.11. Código empleado para obtener los valores en Klux.....	17
3.11.1. Código empleado para obtener los valores en Lux.....	17
3.11.2. Código del programa de interfaz gráfico en Matlab	18
3.13. Proceso de Calibración	30
IMPORTANTE:	31
Conclusiones y recomendaciones	32
Referencias	33

Índice de tablas

Tabla 1	3
Tabla 2	8

Índice de figuras

Figura 1. Estado del Luxómetro	4
Figura 2. Fuente de alimentación (Vieja)	4
Figura 3. Interior del Luxómetro.	5
Figura 4. Placa de circuito Impreso (Antes)	6
Figura 5. Carcasa vacía.....	6
Figura 6. Placa Electrónica del Display.....	7
Figura 7. Placa de circuito Impreso (Después).....	7
Figura 8. Potenciómetro dañado.....	9
Figura 9. Potenciómetro nuevo.....	9
Figura 10. Revisión de componentes de la PCB con Multímetro.	9
Figura 11. Sensor LDR.....	10
Figura 12. Microcontrolador PIC 16F877	11
Figura 13. Pantalla GLCD	11
Figura 14. Placa del Display	11
Figura 15. Fuente de alimentación (Nueva)	12
Figura 16. Nivel de Burbuja.	12
Figura 17. Interruptor	13
Figura 18. Cono impreso en 3D	13
Figura 19. Resortes nuevos.....	14
Figura 20. Display funcional.	15
Figura 21. Respuesta del LDR a la luz	15
Figura 22. Elementos utilizados en el circuito Software Proteus	15
Figura 23 Conexión de los dispositivos.....	16
Figura 24. Simulación del Circuito, Fuente: Programa Proteus	16
Figura 25. Diseño del circuito impreso PCB.....	17
Figura 26. Distancia faro-luxómetro.	30
Figura 27. Distancia H.	30
Figura 28. Rueda de Calibración.....	31

1. Introducción

La correcta calibración de las luces de los vehículos es fundamental para garantizar la seguridad vial, ya que un sistema de iluminación mal ajustado puede provocar deslumbramientos a otros conductores o una iluminación insuficiente de la vía. Para llevar a cabo dicha calibración, es indispensable contar con instrumentos de medición precisos, como el luxómetro, cuya función es medir la intensidad de la luz en unidades de lux. Sin embargo, con el uso prolongado y la exposición a diversas condiciones ambientales, estos dispositivos pueden sufrir daños que afectan la exactitud de sus mediciones.

El presente proyecto se enfoca en la reparación y calibración de un luxómetro dañado con el objetivo de restaurar su funcionalidad. La intervención del instrumento implicó un diagnóstico exhaustivo para identificar fallas en componentes electrónicos clave, como el sensor de luz, la tarjeta de procesamiento y la fuente de alimentación. Se realizaron reemplazos y ajustes técnicos empleando herramientas de precisión y procedimientos normalizados para asegurar la integridad del equipo.

La calibración posterior fue ejecutada en un entorno controlado, utilizando fuentes de luz patrón certificadas y comparando las mediciones del luxómetro con las referencias establecidas. Este proceso permitió verificar que el margen de error se mantuviera dentro de los límites aceptados por las normativas mencionadas, garantizando así mediciones confiables y repetibles.

Este trabajo no solo tiene relevancia técnica al recuperar la operatividad de un equipo de medición especializado, sino que también contribuye a la seguridad vial al permitir la correcta regulación del sistema de iluminación automotriz. Además, promueve el uso responsable de instrumentos de medición y la importancia de su mantenimiento preventivo para prolongar su vida útil y optimizar recursos.

2. Objetivos

2.1 Objetivo General.

- ❖ Reparar un luxómetro diseñado para la calibración de luces de vehículos.

2.2 Objetivos Específicos.

- ❖ Diagnosticar las fallas técnicas del luxómetro y determinar las causas de su mal funcionamiento.
- ❖ Realizar la reparación del luxómetro, asegurando que todos sus componentes operen correctamente.
- ❖ Evaluar el desempeño del luxómetro reparado mediante pruebas prácticas.

3. Procedimiento

Tabla 1

Herramientas	Utilidad en el Proyecto
Multímetro	Permite medir voltajes, corrientes y resistencias para diagnosticar fallas en circuitos eléctricos.
Lupa	Facilita la inspección visual de componentes pequeños y soldaduras.
Cautín	Se usa para soldar componentes electrónicos en placas de circuito impreso.
Estaño	Material conductor utilizado en la soldadura de componentes electrónicos.
Pasta de soldadura	Mejora la adherencia del estaño y facilita la soldadura en los componentes.
Alcohol isopropílico	Se emplea para limpiar residuos de soldadura y suciedad en circuitos electrónicos.
Cepillo pequeño	Ayuda a remover polvo y residuos en circuitos y componentes electrónicos.
Pistola de silicona	Se utiliza para fijar componentes o aislar partes de un circuito.
Cinta termo encogible	Proporciona aislamiento y protección a las conexiones eléctricas.
Playo	Se usa para sujetar o cortar cables y otros materiales en el montaje.
Desarmador	Herramienta básica para desmontar y ensamblar dispositivos electrónicos.

3.1. Estado inicial del luxómetro

- Contaminación generalizada:

Acumulación de polvo en toda la superficie display y carcasa (Figura 1).

Depósitos de grasa/óxido en conectores metálicos.



Figura 1. Estado del Luxómetro

- Daños eléctricos críticos:

Fuente de alimentación: Cable de entrada cortado. El módulo convertidor AC/DC (120VAC a 12VDC/500mA) (Figura 2)



Figura 2. Fuente de alimentación (Vieja)

3.2. Desarme e Identificación de Componentes

El proceso de desarme tuvo como finalidad:

- Acceder a la placa de control y componentes internos para diagnóstico detallado.
- Retirar piezas dañadas o contaminadas (fuente, LDR, componentes sueltos).
- Preparar la carcasa y partes mecánicas para limpieza profunda.

3.2.1. Retiro de tornillos y tapas:

- Se identificaron y removieron los tornillos de la tapa del luxómetro (Figura 3)
- Observación: Algunos tornillos presentaban óxido, lo que dificultó su extracción.



Figura 3. Interior del Luxómetro.

3.2.2. Desconexión de cables y componentes mecánicos:

- Se desconectó el cable plano del display (conector ZIF de 20 pines).
- Se retiraron los cables de alimentación y todo lo que pudiera interferir para la extracción de la PCB (Placa de circuito Impreso)

3.2.3. Extracción de la PCB:

- La placa estaba fijada con 4 tornillos. (Figura 4)

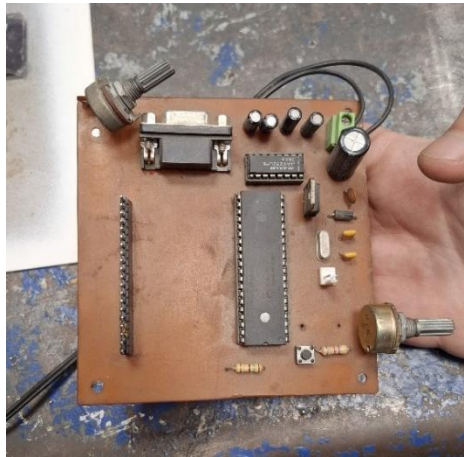


Figura 4. Placa de circuito Impreso (Antes)

3.3. Limpieza General.

Antes de abordar la limpieza de la PCB, se realizó una descontaminación completa de todos los componentes mecánicos y la carcasa, asegurando un entorno libre de polvo y residuos para el reensamblaje posterior.

3.3.1. Limpieza de la carcasa vacía:

- Se utilizó aire comprimido para eliminar polvo superficial y con ayuda de guaípe se limpió la luna de la caja por donde recibirá la intensidad de luz de los halógenos. (Figura 5)
- Resultado: Superficie libre de contaminantes y lista para reensamblaje.

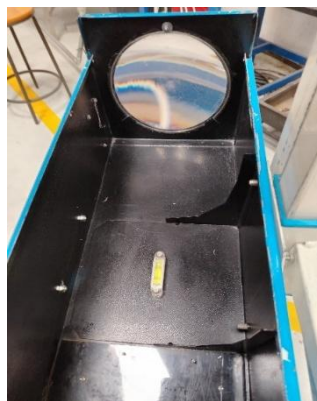


Figura 5. Carcasa vacía.

3.3.2. Limpieza de componentes:

Con ayuda de aire comprimido y guaípe se limpió todo el polvo superficial de los componentes mecánicos que ayudan en el buen funcionamiento del luxómetro (resortes, pernos, etc.)

Tras asegurar que la estructura mecánica estaba libre de contaminantes, se procedió a la limpieza especializada de la placa del display (Figura 6), donde se concentraban los componentes electrónicos críticos.

Display: Se limpio con alcohol isopropílico para no dañar componentes ni el cristal.



Figura 6. Placa Electrónica del Display

3.3.3. Limpieza de la PCB

Remoción de polvo:

- Se utilizó aire comprimido para eliminar polvo superficial.
- Limpieza con alcohol isopropílico:
- Aplicado con cepillo en zonas críticas (conectores, pads de soldadura).

Resultado: Eliminación de residuos conductores, suciedad y óxido (Figura 7).

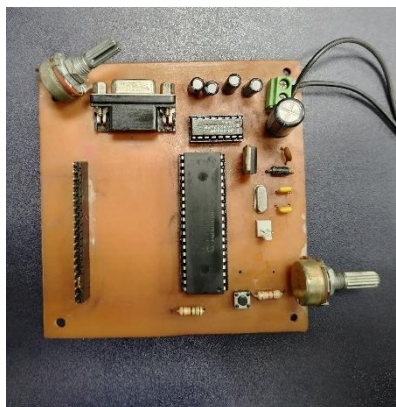


Figura 7. Placa de circuito Impreso (Después)

3.4. Análisis de Componentes Electrónicos de la PCB

3.4.1. Problemas identificados durante la limpieza:

Tabla 2

Componente	Descripción del Daño
Potenciómetro	Presenta fallos en la variación de resistencia, impidiendo ajustes precisos.
Sensor de luz LDR	Responde de manera incorrecta a los cambios de luz, posiblemente debido a envejecimiento o daño interno.
Resorte de regulación	Pérdida de elasticidad o deformación, afectando la calibración del sistema.
Fuente de alimentación	No suministra el voltaje adecuado, posiblemente debido a condensadores en mal estado o cortocircuitos.
Condensadores	Fugas o hinchazón debido a sobrecargas o envejecimiento, afectando el rendimiento del circuito.
Resistencias	Valores fuera de especificación o abiertas, causando mal funcionamiento en los circuitos asociados.
Compuerta lógica MAX2320PE 7846	Fallos en la transmisión de señales, afectando la comunicación entre componentes.

➤ **Potenciómetro roto:**

- Eje fracturado y resistencia irregular (Figura 8)
- Solución: Reemplazo por un modelo nuevo de $5k\Omega$ (Figura 9).



Figura 8. Potenciómetro dañado.

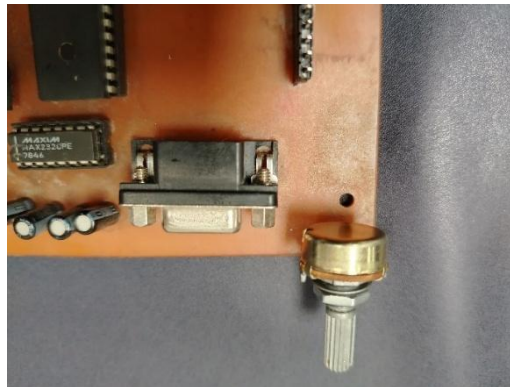


Figura 9. Potenciómetro nuevo

3.4.2. Revisión de los componentes de la PCB.

Garantizar que todos los componentes electrónicos de la PCB, excepto los ya identificados como dañados (potenciómetro, LDR y fuente), se encontraban funcionales y dentro de sus parámetros técnicos (Figura 10).

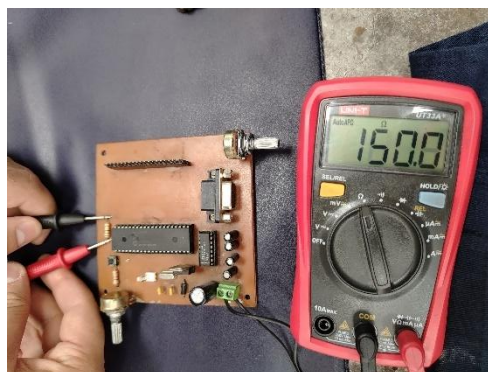


Figura 10. Revisión de componentes de la PCB con Multímetro.

3.5. Componentes e Identificación que integran el Luxómetro.

Se identificó los componentes internos específicamente en la parte electrónica, el cual tiene:

- Sensor de Luminosidad LDR (Light Dependent Resistance), este dispositivo cuya resistencia óhmica varía en función de la iluminación recibida sobre una superficie (Figura 11)
- Microcontrolador PIC 16F877, la cual permite realizar la conversión Analógica / Digital con una resolución de 10 bits (Figura 12)
- Pantalla GLCD (Graphical Led Display), el cual sirve para visualizar los valores digitales que se obtienen mediante el ADC del PIC (Figura 13)
- Placa del Display (Figura 14).
- Potenciómetros de 5k Ω
- Condensadores
- Resistencias
- Diodos
- Fuente de poder 12VDC 1A (Figura 15)
- Nivel de burbuja (Figura 16).

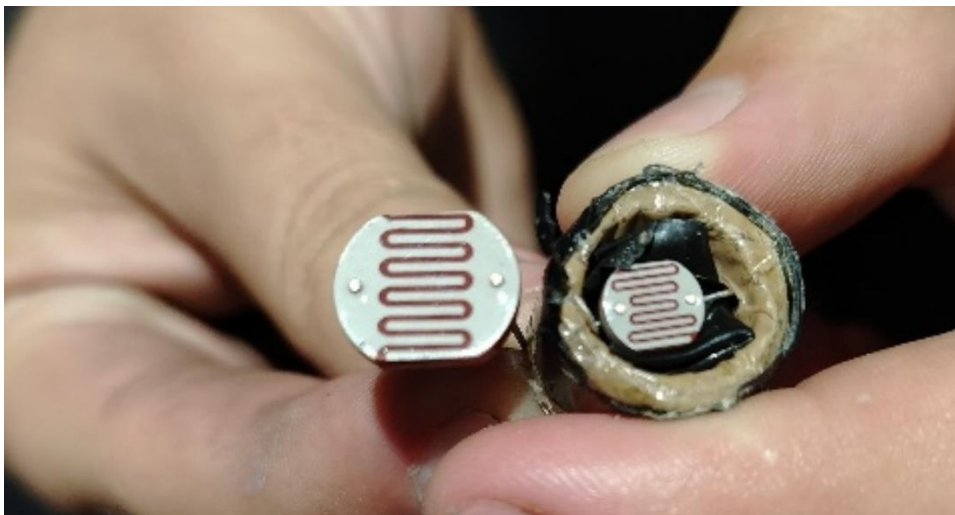


Figura 11. Sensor LDR



Figura 12. Microcontrolador PIC 16F877



Figura 13. Pantalla GLCD



Figura 14. Placa del Display



Figura 15. Fuente de alimentación (Nueva)



Figura 16. Nivel de Burbuja.

3.6. Montaje del luxómetro

Una vez completada la limpieza exhaustiva de todos los componentes y el reemplazo de las piezas dañadas (LDR, potenciómetro, fuente de alimentación y diodo), se procedió al montaje del equipo:

3.6.1. Montaje de la PCB reparada:

- Se instaló la placa de circuito impreso (PCB) en la carcasa previamente limpiada, asegurando una fijación estable con los soportes y tornillos correspondientes.

3.6.2. Montaje de los componentes Mecánicos:

- Se instaló todos los componentes mecánicos asegurando de que nada quedara flojo que después pudiera afectar en el rendimiento del luxómetro.

3.6.3. Mejoras realizadas

- Se instaló un interruptor para el paso del corriente anclado a la estructura, ya que anteriormente no era funcional ya que se tenía corriente directa. (Figura 17)



Figura 17. Interruptor

- Se construyó un cono hecho en 3D con el propósito de dirigir la luz directamente hacia el LDR, evitando la dispersión del haz y obteniendo así mediciones más precisas (Figura 18)



Figura 18. Cono impreso en 3D

- Instalación de nuevos resortes para el buen funcionamiento de la regulación en altura (Figura 19)



Figura 19. Resortes nuevos

3.6.4. Conexión de componentes:

- Se instaló los nuevos cables de alimentación al nuevo módulo de fuente de 12V.
- Se conectó el display y el nuevo LDR, verificando polaridad y continuidad.

3.6.5. Ajuste del potenciómetro:

- Se calibró manualmente el nuevo potenciómetro para asegurar un rango de medición óptimo.

3.6.6. Colocación de tapa:

- Se colocó la tapa asegurándonos de conectar el zif de 20 pines al display para su correcto funcionamiento.

3.7. Prueba de Encendido y Medición.

Al alimentar el equipo, se observó:

- Display funcional: Mostró valores numéricos estables en lux (Figura 20).
- Respuesta del LDR: Al exponer el sensor a diferentes fuentes de luz (linterna, luz ambiente, faro de vehículo), los valores variaron acorde a la intensidad lumínica (Figura 21)



Figura 20. Display funcional.



Figura 21. Respuesta del LDR a la luz

3.8. Preparación de placa electrónica nueva

Para la elaboración del circuito se siguió el siguiente proceso:

Para garantizar el correcto funcionamiento del diseño, se realizó una simulación virtual del circuito electrónico utilizando el software especializado *Proteus ISIS*. Esta herramienta cuenta con una amplia librería de componentes que permite emular el comportamiento del sistema en tiempo real, facilitando la detección temprana de posibles errores. (Figura 22).

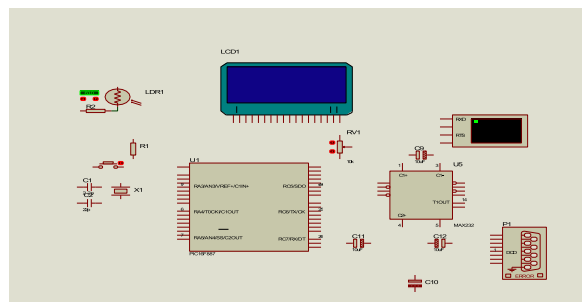


Figura 22. Elementos utilizados en el circuito Software Proteus

Fuente: Programa Proteus.

Se realizó el interconectado de los componentes electrónicos siguiendo cuidadosamente las especificaciones de los pines del microcontrolador y la pantalla LCD. Para garantizar la correcta implementación, se consultaron minuciosamente las hojas técnicas (datasheets) de cada circuito integrado, identificando claramente: los pines de alimentación (Vcc), tierra (GND), así como aquellos destinados a funciones específicas de comunicación y control. Este proceso meticuloso aseguró que todas las conexiones cumplieran con los requerimientos técnicos del diseño. (Figura 23).

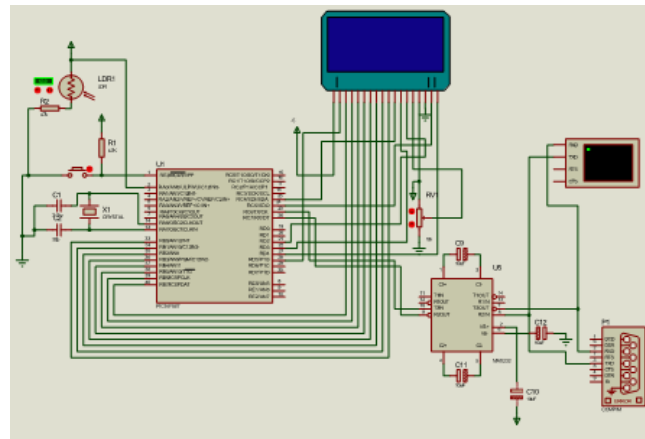


Figura 23 Conexión de los dispositivos

Fuente: Programa Proteus.

Se procedió a cargar el firmware desarrollado en MikroC al microcontrolador, configurando previamente la frecuencia de operación para que coincidiera exactamente con los parámetros establecidos en el entorno de programación. Durante la ejecución del programa, es importante mencionar que cualquier modificación realizada en el código fuente se reflejará inmediatamente en la simulación de Proteus, excepto cuando el simulador se encuentre en modo de ejecución activa. (Figura 24).

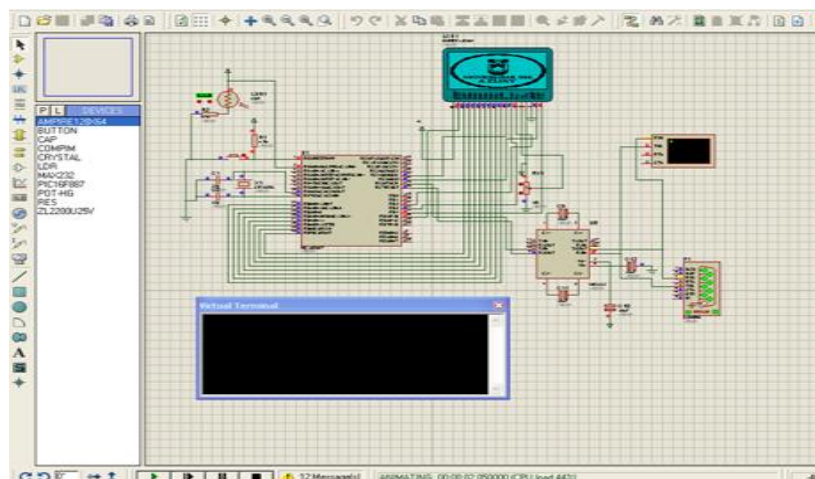


Figura 24. Simulación del Circuito, Fuente: Programa Proteus

Tras confirmar el óptimo funcionamiento del circuito mediante simulación, se procedió al diseño de la placa de circuito impreso (PCB) utilizando el módulo Ares de Proteus ISIS. Esta herramienta permite la conversión automática del esquemático simulado a diseño de PCB, manteniendo intacta la topología de conexiones establecida. (Figura 25).

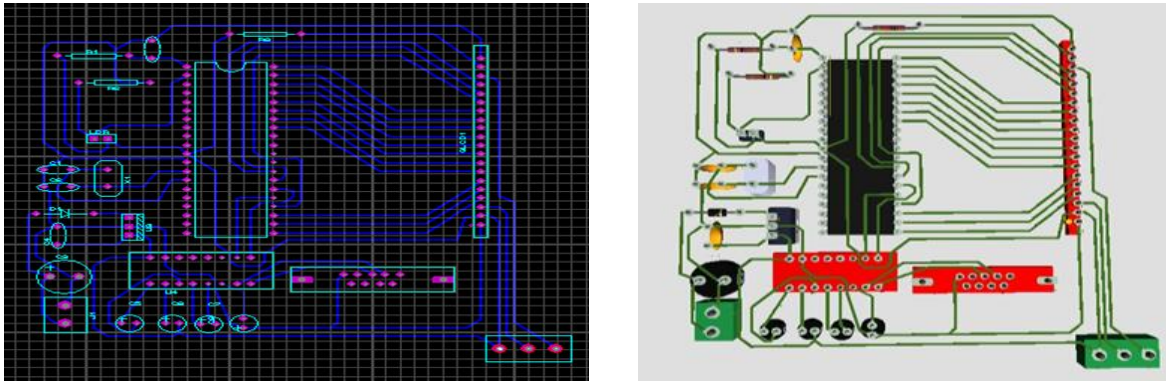


Figura 25. Diseño del circuito impreso PCB

Fuente: Programa Proteus.

3.11. Código empleado para obtener los valores en Klux

```
global com_ser

global K_lux while true

B= (fscanf(com_ser,'%f',5)./1000); disp(B); set(handles.K_lux,'string',B)

pause (0.2); end
```

3.11.1. Código empleado para obtener los valores en Lux

```
global com_ser

global text_5

%A= [];

while true pause(0.2);

%for j= 1:10

A= fscanf(com_ser,'%f',5); disp(A); set(handles.text5,'string',A); end
```

3.11.2. Código del programa de interfaz gráfico en Matlab

```
function varargout = rx_sensor(varargin)

% RX_SENSOR M-file for rx_sensor.fig

%     RX_SENSOR, by itself, creates a new RX_SENSOR or raises the existing
%     singleton*.

%
%     H = RX_SENSOR returns the handle to a new RX_SENSOR or the handle to
%     the existing singleton*.

%
%     RX_SENSOR('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in RX_SENSOR.M with the given input arguments.

%
%     RX_SENSOR('Property','Value',...) creates a new RX_SENSOR or raises the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before rx_sensor_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to rx_sensor_OpeningFcn via varargin.

%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%     instance to run (singleton)".

%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help rx_sensor

% Last Modified by GUIDE v2.5 27-Mar-2012 14:31:38
```

```

% Begin initialization code - DO NOT EDIT gui_Singleton = 1;

gui_State = struct('gui_Name', mfilename, ... 'gui_Singleton', gui_Singleton, ...
'gui_OpeningFcn', @rx_sensor_OpeningFcn, ... 'gui_OutputFcn', @rx_sensor_OutputFcn, ...
'gui_LayoutFcn', [] , ...
'gui_Callback', []); if nargin && ischar(varargin{1})
gui_State.gui_Callback = str2func(varargin{1}); end

if nargout
[varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
gui_mainfcn(gui_State, varargin{:}); end

% End initialization code - DO NOT EDIT

% --- Executes just before rx_sensor is made visible.

function rx_sensor_OpeningFcn(hObject, eventdata, handles, varargin)

% This function has no output args, see OutputFcn.

% hObject    handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles     structure with handles and user data (see GUIDATA)

% varargin command line arguments to rx_sensor (see VARARGIN) set(handles.portList, 'String',
...

['COM1']; ...

['COM2']; ...

['COM3']; ...

['COM4']; ...

['COM5']; ...

['COM6']; ...

['COM7']; ...

```

```

{'COM8'}; ...

{'COM9'}; ...

{'COM10'}; ...

{'COM11'}; ...

{'COM12'}; ...

{'COM13'}; ...

{'COM14'} ]]);

set(handles.portList, 'Value', 2);

% Choose default command line output for rx_sensor handles.output = hObject;

% Update handles structure global linea

global linea1 global linea2 global linea3 global linea4 global linea5 linea='1'; linea1='1';
linea2='1'; linea3='1';

linea4='1'; linea5='1';

guidata(hObject, handles);

% UIWAIT makes rx_sensor wait for user response (see UIRESUME)

% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line. function varargout =
rx_sensor_OutputFcn(hObject, eventdata, handles)

% varargout cell array for returning output args (see VARARGOUT);

% hObject      handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure varargout{1} = handles.output;

function KLUX_Callback(hObject, eventdata, handles)

% hObject      handle to KLUX (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB

% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of KLUX as text

%      str2double(get(hObject,'String')) returns contents of KLUX as a double global com_ser
global K_lux while true

B= (fscanf(com_ser,'%f',5)./1000); disp(B); set(handles.K_lux,'string',B)

pause (0.2); end

%%K1= [];

%while (1==1)

%K1 = [K1 fread(com_ser,1,'uchar')];

%K1=string(K1);

%disp(A);

% --- Executes during object creation, after setting all properties. function
KLUX_CreateFcn(hObject, eventdata, handles)

% hObject      handle to KLUX (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

%      See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white'); end

function LUX_Callback(hObject, eventdata, handles)

% hObject      handle to LUX (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles      structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of LUX as text

%      str2double(get(hObject,'String')) returns contents of LUX as a double global com_ser

global text_5

%A= [];

while true pause(0.2);

%for j= 1:10

A= fscanf(com_ser,'%f',5); disp(A); set(handles.text5,'string',A); end;

%B= (fscanf(com_ser,'%f',5)./1000);

%disp(B);

%set(handles.K_lux,'string',B)

%set(handles.K_lux,'string',A);

%end

%K1= [];

%while (1==1)

%K1 = [K1 fread(com_ser,1,'uchar')];

%K1=string(K1);

%pause(1)

%end;

% --- Executes during object creation, after setting all properties. function
LUX_CreateFcn(hObject, eventdata, handles)

% hObject      handle to LUX (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

%      See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white'); end

% --- Executes on button press in conect.

function conect_Callback(hObject, eventdata, handles)

% hObject      handle to conect (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles      structure with handles and user data (see GUIDATA) clc

global com_ser

if strcmp(get(hObject,'String'),'Connect') serPortn = get(handles.portList, 'Value'); serList =
get(handles.portList,'String'); serPort = serList{serPortn};

%puerto= get(handles.PORT,'string');

%com_ser=serial(puerto);    com_ser=    serial(serPort);    set(com_ser,'BaudRate',9600);
set(com_ser,'DataBits',8);    set(com_ser,'Parity','none');    set(com_ser,'StopBits',1);
set(com_ser,'FlowControl','none');    set(com_ser,'Terminator','CR/LF');
set(com_ser,'InputBufferSize',20);

try fopen(com_ser);

com_ser = com_ser; set(handles.rxBUTTON, 'Enable', 'On'); set(hObject, 'String','Disconnect')
catch e

errordlg(e.message); end

else

%set(handles.Tx_send, 'Enable', 'Off'); set(handles.rxBUTTON, 'Enable', 'Off');

set(hObject, 'String','Connect') fclose(com_ser); delete(com_ser);

clear com_ser; end

INSTRFIND

guidata(hObject, handles);

%set(com_ser,'Timeout',5);

```

```

%fopen(com_ser);

% --- Executes when user attempts to close figure1.

function figure1_CloseRequestFcn(hObject, eventdata, handles)

% hObject      handle to figure1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles      structure with handles and user data (see GUIDATA) global com_ser;

%fclose(com_ser); delete(com_ser); clear com_ser;

% Hint: delete(hObject) closes the figure delete(hObject);

% --- Executes on button press in alt_der.

function alt_der_Callback(hObject, eventdata, handles)

% hObject      handle to alt_der (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles      structure with handles and user data (see GUIDATA) global K_lux

global linea lin=str2num(linea); linea=lin+1; linea=num2str(linea);

a=warndlg('GUARDANDO DATO','GUARDANDO');

Luz_dera=get(handles.K_lux,'string');    Luz_dera=str2num(Luz_dera);    cadena=['A',linea];
xlswrite('reglas.xls',Luz_dera,'DATOS',cadena); close(a);

warndlg('DATO GUARDADO','END');

guidata(hObject,handles);

% --- Executes on button press in alt_izq.

function alt_izq_Callback(hObject, eventdata, handles)

% hObject      handle to alt_izq (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles      structure with handles and user data (see GUIDATA)

%handles.q=inputdlg('INGRESE EL VALOR MEDIDO','GUARDAR: ');

```

```

global K_lux global linea1

lin1=str2num(linea1);

linea1=lin1+1; linea1=num2str(linea1);

a=warndlg('GUARDANDO DATO','GUARDANDO');

Luz_iza=get(handles.K_lux,'string');      Luz_iza=str2num(Luz_iza);      cadena=['B',linea1];
xlswrite('reglas.xls',Luz_iza,'DATOS',cadena); close(a);

warndlg('DATO GUARDADO','END');

guidata(hObject,handles);

% --- Executes on button press in open.

function open_Callback(hObject, eventdata, handles)

% hObject      handle to open (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles      structure with handles and user data (see GUIDATA)

%[FileName Path]=uigetfile({'*.doc;*.xls'},'ABRIR DOCUMENTO');

%if isequal(FileName,0)

% return

%else

%winopen(strcat(Path,FileName));

%winopen('D:/myfiles/myresults.html') winopen('reglas_car.xls')

%datos=load('datos.txt');

%end

% --- Executes on button press in baj_iz.

function baj_iz_Callback(hObject, eventdata, handles)

% hObject      handle to baj_iz (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

```

```

% handles      structure with handles and user data (see GUIDATA) global K_lux

global linea2 lin2=str2num(linea2); linea2=lin2+1; linea2=num2str(linea2);

a=warndlg('GUARDANDO DATO','GUARDANDO');

Luz_bajiz=get(handles.K_lux,'string'); Luz_bajiz=str2num(Luz_bajiz); cadena=['C',linea2];
xlswrite('reglas.xls',Luz_bajiz,'DATOS',cadena); close(a);

warndlg('DATO GUARDADO','END');

guidata(hObject,handles);

% --- Executes on button press in baj_der.

function baj_der_Callback(hObject, eventdata, handles)

% hObject      handle to baj_der (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles      structure with handles and user data (see GUIDATA)

global K_lux global linea3

lin3=str2num(linea3); linea3=lin3+1; linea3=num2str(linea3);

a=warndlg('GUARDANDO DATO','GUARDANDO');

Luz_bajder=get(handles.K_lux,'string'); Luz_bajder=str2num(Luz_bajder); cadena=['D',linea3];
xlswrite('reglas.xls',Luz_bajder,'DATOS',cadena); close(a);

warndlg('DATO GUARDADO','END');

guidata(hObject,handles);

% --- Executes on button press in halo_iz.

function halo_iz_Callback(hObject, eventdata, handles)

% hObject      handle to halo_iz (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles      structure with handles and user data (see GUIDATA) global K_lux

global linea4 lin4=str2num(linea4); linea4=lin4+1; linea4=num2str(linea4);

```

```

a=warndlg('GUARDANDO DATO','GUARDANDO');

Luz_haloiz=get(handles.K_lux,'string'); Luz_haloiz=str2num(Luz_haloiz); cadena=['E',linea4];
xlswrite('reglas.xls',Luz_haloiz,'DATOS',cadena); close(a);

warndlg('DATO GUARDADO','END');

guidata(hObject,handles);

% --- Executes on button press in halo_der.

function halo_der_Callback(hObject, eventdata, handles)

% hObject      handle to halo_der (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles      structure with handles and user data (see GUIDATA)

global K_lux global linea5

lin5=str2num(linea5); linea5=lin5+1; linea5=num2str(linea5);

a=warndlg('GUARDANDO DATO','GUARDANDO');

Luz_haloder=get(handles.K_lux,'string'); Luz_haloder=str2num(Luz_haloder);
cadena=['F',linea5]; xlswrite('reglas.xls',Luz_haloder,'DATOS',cadena); close(a);

warndlg('DATO GUARDADO','END');

guidata(hObject,handles);

function Lin_excel_Callback(hObject, eventdata, handles)

% hObject      handle to Lin_excel (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Lin_excel as text

%      str2double(get(hObject,'String')) returns contents of Lin_excel as a double

% --- Executes during object creation, after setting all properties. function
Lin_excel_CreateFcn(hObject, eventdata, handles)

% hObject      handle to Lin_excel (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB

% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

%      See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white'); end

% --- Executes during object creation, after setting all properties. function
fig1_CreateFcn(hObject, eventdata, handles)

% hObject      handle to fig1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles      empty - handles not created until after all CreateFcns called

[x,map]=imread('U_Azuay.jpg','jpg'); image(x),colormap(map),axis off,hold on

% Hint: place code in OpeningFcn to populate fig1

% --- Executes on button press in salir.

function salir_Callback(hObject, eventdata, handles)

% hObject      handle to salir (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles      structure with handles and user data (see GUIDATA)

opc=questdlg('¿Salir De Control de Luminosidad?','SALIR','Si','No','No'); if strcmp(opc,'No')

return; fclose(com_ser);

delete(com_ser); clear com_ser;

end clear,clc,close all

% --- Executes on selection change in portList.

function portList_Callback(hObject, eventdata, handles)

% hObject      handle to portList (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB

% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns portList contents as cell array

%      contents{get(hObject,'Value')} returns selected item from portList

% --- Executes during object creation, after setting all properties. function
portList_CreateFcn(hObject, eventdata, handles)

% hObject      handle to portList (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles      empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.

%      See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white'); end

% --- Executes on button press in rxButton.

function rxButton_Callback(hObject, eventdata, handles)

% hObject      handle to rxButton (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles      structure with handles and user data (see GUIDATA)

```

3.13. Proceso de Calibración

1. Alinear el luxómetro paralelamente al eje longitudinal del vehículo.
2. Tomamos las medidas referenciales:
 - La distancia entre faro y luxómetro puede estar entre (20 y 50) centímetros, teniendo en cuenta una perfecta nivelación de caja óptica y una alineación paralela al vehículo mediante el espejo de precisión tomando una referencia horizontal en el vehículo (Figura 26)



Figura 26. Distancia faro-luxómetro.

- Medimos la altura H desde el centro del faro al suelo, y colocamos al luxómetro en la misma medida de altura ()



- Figura 27)

Figura 27. Distancia H .

IMPORTANTE:

El haz de luz debe incidir directamente sobre el sensor que está en el centro de la placa, para evitar falla en la medida, para esto el centrado entre faro y luxómetro debe ser lo más preciso posible

1. Actuamos sobre la rueda de calibración de acuerdo al tipo de vehículo que se va a verificar (Figura 28)



Figura 28. Rueda de Calibración.

Conclusiones y recomendaciones

Tras dos meses de trabajo, en los que se llevaron análisis para la correcta comprobación y reparación, se logró restaurar por completo la funcionalidad del luxómetro objeto de estudio.

Este proceso no solo representó un reto técnico, sino también una valiosa oportunidad de aprendizaje y desarrollo profesional. A lo largo de este periodo, se identificaron y corrigieron fallas en componentes clave del equipo, lo que permitió extender su vida útil y evitar costos innecesarios en la adquisición de un nuevo instrumento.

Se demostró que el luxómetro cumple con los márgenes de error aceptables para ser utilizado en la calibración de luces vehiculares. Esto cobra especial relevancia si se considera el impacto directo que tiene una correcta calibración de la iluminación automotriz en la seguridad vial. Un luxómetro en óptimas condiciones garantiza que las luces de los vehículos no provoquen deslumbramientos ni dejen zonas de baja visibilidad, lo que contribuye a reducir accidentes en las carreteras.

Con base en la experiencia obtenida y en los resultados alcanzados, se plantean las siguientes recomendaciones para garantizar la continuidad y el óptimo desempeño del luxómetro:

- **Mantenimiento preventivo:** Se sugiere implementar un programa de revisiones periódicas que contemple la limpieza del sensor, la verificación de conexiones internas y la comprobación de la batería o fuente de alimentación. Esto permitirá anticiparse a posibles fallas y prolongar la vida útil del equipo.
- **Calibración periódica:** Se recomienda realizar calibraciones anuales, o bien, después de cualquier reparación o ante la detección de mediciones inconsistentes. Utilizar fuentes de luz patrón certificadas garantizará resultados precisos.
- **Protección del equipo:** Invertir en estuches protectores y condiciones de almacenamiento adecuadas minimizará daños por golpes, polvo o humedad.

Referencias

- ❖ Salazar, M. (2023, 28 de marzo). *Luxómetros: una guía completa para entender, elegir y utilizar estas herramientas de medición de luz*. Sodeintec SAS. <https://sodeintecsas.com/luxometros-una-guia-completa-para-entender-elegir-y-utilizar-estas-herramientas-de-medicion-de-luz/>
- ❖ Leal Importaciones. (2023, 13 de octubre). *Luxómetro: Qué es y su función en la revisión técnica vehicular*. Leal Importaciones. <https://lealimportaciones.com/luxometro-importancia-funcionamiento-crtv/>
- ❖ Cassingena Navone, E. (2022, 24 de febrero). Python ejemplos de código – tutorial de programación en Python desde cero para principiantes. freeCodeCamp. <https://www.freecodecamp.org/espanol/news/python-ejemplos-de-codigo-tutorial-de-programacion-en-python-desde-cero-para-principiantes/>
- ❖ Rodríguez, I. (s.f.). *Programar y simular con Proteus*. Scribd. <https://es.scribd.com/document/189768103/Programar-y-Simular-Con-Proteus>
- ❖ Hubor. (s.f.). *Código fuente (simulación de microprocesadores VSM)*. <https://www.hubor-proteus.com/proteus-pcb/proteus-pcb/16-vsm.html>
- ❖ Viresa. (2023, 10 de junio). *¿Qué es y cómo funciona un luxómetro?* https://viresa.com.mx/blog_funcionamiento_luxometro
- ❖ Faro Barcelona. (s.f.). *¿Cómo se mide la luz? Unidades de medidas y luxómetro*. <https://faro.es/es/blog/como-se-mide-la-luz/>
- ❖ Centro de Diagnóstico Automotor. (2019, 31 de enero). *Revisión técnico mecánica*. <https://www.centrodediagnosticoautomotor.com/2019/01/31/revision-tecnico-mecanica/>

- ❖ Sánchez, M. (2025, 21 de enero). *¿Qué es un microprocesador? ¿Cuál es su función?* PcComponentes. <https://www.pccomponentes.com/que-es-un-microprocesador-cual-es-su-funcion>
- ❖ Transfer Multisort Elektronik. (2024, 25 de marzo). *¿Cómo programar un microcontrolador? Los lenguajes de programación de microcontroladores más populares.* <https://www.tme.eu/es/news/library-articles/page/58200/como-programar-un-microcontrolador-los-lenguajes-de-programacion-de-microcontroladores-mas-populares/>
- ❖ Air Liquide. (s.f.). *¿Cómo se diseñan las placas electrónicas?* <https://es.airliquide.com/soluciones/procesos-con-gas-para-el-ensamblaje-electronico/como-se-disenan-las-placas-electronicas>