

UTN
FACULTAD REGIONAL
VILLA MARÍA

Ingeniería en Sistemas de Información

Teoría de Control

Proyecto de Arduino

Integrantes

- ❖ Anil Agustin
- ❖ Baldassi Gonzalo
- ❖ Bustos Samuel
- ❖ Godoy Santiago
- ❖ Goti Franco
- ❖ Salera Roberto

Profesores:

- ❖ Ing. José Luis Catalano
- ❖ Ing. Nicolas Oliva

Fecha de entrega: 24/11/2023

Índice

Resumen.....	3
Marco Teórico.....	3
Modelado y Funciones de Transferencia:.....	3
Régimen Permanente y Transitorio:.....	4
Controladores:.....	4
Módulo ESP8266:.....	4
Descripción detallada del diseño del sistema.....	5
Desarrollo del Código.....	5
Configuración de WiFi:.....	5
Función para apagar gradualmente el LED:.....	6
Recepción del mensaje.....	8
Esquema de Conexionado.....	9
Conclusiones.....	11

Resumen

En el marco del presente proyecto se ha utilizado una protoboard a la cual se le ha integrado un módulo ESP8266MOD junto con los conectores pertinentes, tales como resistencias, sensores, LEDs, entre otros elementos, con el propósito de configurar un sistema. Este sistema, diseñado con el fin de funcionar como un sensor para el alumbrado público, responde apagando un LED gradualmente a medida que un sensor LDR es iluminado con un láser KY-800. Esta estructura hace analogía a un sistema que regula el alumbrado público. A medida que el sol(láser) va impactando una fotocélula(LDR) la luz va disminuyendo lentamente.

El propósito general de este proyecto consiste en adquirir conocimientos relativos a la aplicación de un módulo ESP8266, en conjunto con otros componentes, con el fin de simular un sistema.

El objetivo específico de la presente iniciativa es lograr la simulación eficaz del funcionamiento de un sistema de alumbrado público, así como alcanzar una transferencia de datos efectiva a través de la conexión WiFi mediante el Packet Sender.

Marco Teórico

Este proyecto puede ser analizado desde la perspectiva de diversos conceptos fundamentales en teoría de control y sistemas dinámicos.

Modelado y Funciones de Transferencia:

Para abordar el comportamiento del sistema, es esencial realizar un modelado que describa la relación entre las entradas (como la iluminación del láser al LDR) y las salidas (activación del LED). Este modelo puede llevarse a cabo mediante funciones de transferencia, que representan la relación matemática entre la entrada y la salida del sistema.

Una función de transferencia es un concepto fundamental en el análisis y diseño de sistemas dinámicos, especialmente en ingeniería de control y teoría de sistemas. Representa la relación matemática entre la entrada y la salida de un sistema dinámico lineal e invariante en el tiempo (LTI). En otras palabras, describe cómo el sistema responde a una entrada específica a lo largo del tiempo.

Régimen Permanente y Transitorio:

El análisis del sistema también involucra la consideración de sus comportamientos en régimen permanente y transitorio. En régimen permanente, se evalúa el estado estable del sistema después de que las condiciones iniciales han desaparecido. Por otro lado, el análisis transitorio se centra en cómo el sistema evoluciona desde condiciones iniciales hasta el régimen permanente después de una perturbación o cambio en la entrada.

En este caso, en régimen permanente, el sistema se encuentra en un estado estable cuando no hay obstrucción de la luz láser, mientras que el análisis transitorio se aplica cuando ocurre la obstrucción, desvaneciendo el LED.

Controladores:

El concepto de controladores es crucial para entender cómo se puede modular y optimizar el comportamiento del sistema. Un controlador podría ser implementado para ajustar la sensibilidad del sistema a la recepción de la luz láser para regular la desactivación del LED.

Módulo ESP8266:

El módulo ESP8266 es un dispositivo de conectividad inalámbrica muy popular que permite la conexión a redes Wi-Fi.

Incluye un procesador integrado que permite la ejecución de código. Esto permite que el módulo funcione de manera autónoma y realice tareas de procesamiento. Viene con memoria flash integrada para almacenar el firmware y los programas que se ejecutan en el dispositivo.

Una de las razones de su popularidad es su relativo bajo costo, lo que lo hace accesible para proyectos de electrónica de bajo presupuesto.

Este módulo se ha convertido en una opción popular para proyectos DIY ("hazlo tú mismo" o "do it yourself" por sus siglas en inglés), prototipos y desarrollos debido a su facilidad de uso, versatilidad y costo asequible. Además, hay una comunidad activa de desarrolladores que comparten proyectos, código y recursos relacionados con el ESP8266.

Datos técnicos:

- Voltaje de operación entre 3V y 3,6V
- Corriente de operación 80 mA
- Temperatura de operación -40°C y 125°C

Descripción detallada del diseño del sistema

Existe un módulo denominado ESP8266MOD que opera como una suerte de plataforma Arduino, siendo empleado para integrar todos los componentes electrónicos necesarios para el funcionamiento del dispositivo. Se conectan a los puertos del módulo una LDR (resistencia foto variable), un LED y un relé mediante señales analógicas. Asimismo, se incorpora el dispositivo láser KY-800, el cual simula la emisión de luz solar.

La LDR se conecta al extremo positivo de la señal, mientras que su extremo negativo se enlaza a la placa junto con una resistencia para regular la señal analógica recibida. En caso de que el valor analógico supere cierto umbral, se activa la lógica del sistema.

El LED, por su parte, conecta su ánodo a la referencia y su cátodo al pin correspondiente del módulo, junto con una resistencia. Este LED simula un poste de luz del alumbrado público. Cuando la luz incide directamente sobre el sensor LDR, el LED se atenúa gradualmente hasta alcanzar un nivel mínimo. Al acercarse al nivel mínimo, se activa el relé para indicar que la luz está momentáneamente apagada, asumiendo que el entorno se encuentra en una fase diurna.

Al retirar el láser del sensor LDR, el LED se ilumina progresivamente, simulando el ocaso del sol y la disminución de la iluminación.

De manera simultánea, se utiliza la aplicación Packet Sender como un verificador. Desde el dispositivo ESP se envía un mensaje a través de una red WiFi indicando que el LED se está apagando. Tras un periodo de tiempo determinado (debido a la latencia de la conexión realizada), la aplicación devuelve un mensaje al dispositivo ESP notificando un cambio de estado en el LED.

Desarrollo del Código

Fragmentos claves del código desarrollado para el funcionamiento del sistema.

Configuración de WiFi:

Se establece el nombre de la red y la contraseña en las variables ssid y password.

```
// Parámetros de WiFi a configurar
const char *ssid = "Fibertel 2.4";    // Nombre de la red WiFi
const char *password = "Salera2812";  // Contraseña de la red WiFi
```

```

void setup(void) {
  pinMode(laserPin, OUTPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(relayPin, OUTPUT);

  Serial.begin(9600);

  // Conexión a La red WiFi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println(WiFi.localIP());

  // Inicio del servidor
  Serial.println("Server ON");
  server.begin();
}

```

Definición de pines:

Se asignan los pines de la placa a las variables laserPin, ledPin, relayPin, y ldrPin para controlar el láser, el LED, el relé y el sensor LDR respectivamente.

```

// Pines de conexión de los componentes electrónicos
const int ledPin = 5;           // Pin para controlar el LED
const int relayPin = 4;         // Pin para controlar el relé
const int ldrPin = A0;          // Pin para la lectura del sensor LDR

```

Loop principal:

La función loop() en este código se encarga de la gestión continua del sistema, la conexión WiFi, el control de componentes electrónicos y la comunicación con servidores remotos y locales. Este bucle se ejecuta repetidamente mientras el dispositivo esté encendido.

```

void loop() {

```

Función para apagar gradualmente el LED:

La función utiliza un bucle para disminuir gradualmente el brillo del LED desde su nivel actual hasta 0 en un periodo de 5 segundos. Además, hay dos ajustes adicionales:

- Si el sensor LDR sigue activado y han pasado al menos 2 segundos desde el inicio de la lectura, el brillo se reduce a la mitad.
- Si el sensor LDR sigue activado y han pasado al menos 4 segundos desde el inicio de la lectura, el LED se apaga completamente (brillo 0).

Estos ajustes buscan proporcionar una transición suave y adaptativa en función del tiempo transcurrido y las condiciones del entorno detectadas por el sensor LDR.

```
// Función para apagar gradualmente el LED
void fadeOutLED() {
    unsigned long startTime = millis();
    unsigned long elapsedTime = 0;
    int startBrightness = lastBrightness;

    while (elapsedTime < 5000 && lastLDRState) {
        int brightness = map(elapsedTime, 0, 5000, startBrightness, 0);
        analogWrite(ledPin, brightness);
        delay(10);
        elapsedTime = millis() - startTime;
    }

    if (lastLDRState && (millis() - ldrStartTime >= 2000)) {
        int halfBrightness = startBrightness / 2;
        analogWrite(ledPin, halfBrightness);
        lastBrightness = halfBrightness;
    }

    if (lastLDRState && (millis() - ldrStartTime >= 4000)) {
        analogWrite(ledPin, 0);
        lastBrightness = 0;
    }
}
```

Función para prender gradualmente el LED:

La función utiliza un bucle for para aumentar progresivamente el brillo del LED desde 0 hasta 255 (el valor máximo permitido por la función analogWrite() en plataformas como Arduino). Cada iteración del bucle incrementa el brillo en 5 unidades y agrega una pausa de 20 milisegundos para suavizar la transición entre los niveles de brillo.

Después de que el bucle se completa y el brillo alcanza su máximo (255), la función actualiza la variable lastBrightness con el valor máximo, lo que indica que el LED está ahora encendido al máximo brillo. Esta función se utiliza al inicio del programa (setup()) para proporcionar una transición suave al encender el LED.

```
// Función para encender gradualmente el LED
void fadeInLED() {
    for (int brightness = 0; brightness <= 255; brightness += 5) {
        analogWrite(ledPin, brightness);
        delay(20);
    }
    lastBrightness = 255;
}
```

Envío del mensaje

Este fragmento de código se encarga de enviar el mensaje "Encendido - Enciende LED" a un servidor remoto con la dirección IP "192.168.0.83" y el puerto 3000, siempre y

cuando el dispositivo ESP8266 esté conectado a la red WiFi. Este tipo de comunicación se utiliza comúnmente para enviar información o comandos entre dispositivos a través de una red.

```
// Envío de mensaje al servidor remoto
String mensaje = ("Encendido - Enciende LED");
const char *mensajeChar = mensaje.c_str();

if (WiFi.status() == WL_CONNECTED) {
  WiFiClient client;
  if (client.connect("192.168.0.83", 3000)) {
    client.write(mensajeChar);
  }
}
```

Recepción del mensaje

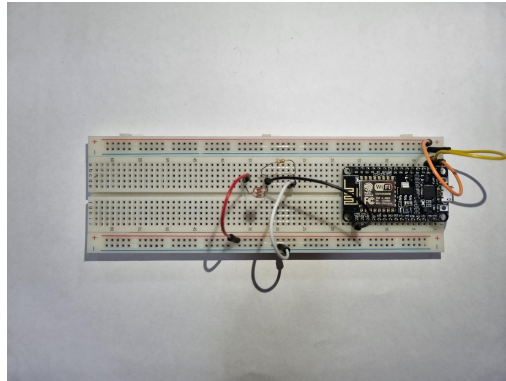
En este fragmento de código se está leyendo la respuesta del servidor remoto caracter por caracter, construyendo así una cadena de respuesta completa. Posteriormente, esa respuesta se imprime en el puerto serie para su visualización y depuración. Este proceso es útil para verificar la respuesta del servidor y entender cómo ha procesado el mensaje enviado por el dispositivo ESP8266.

```
// Espera y lee la respuesta del servidor
String response;
while (client.available()) {
  char c = client.read();
  response += c ;
}

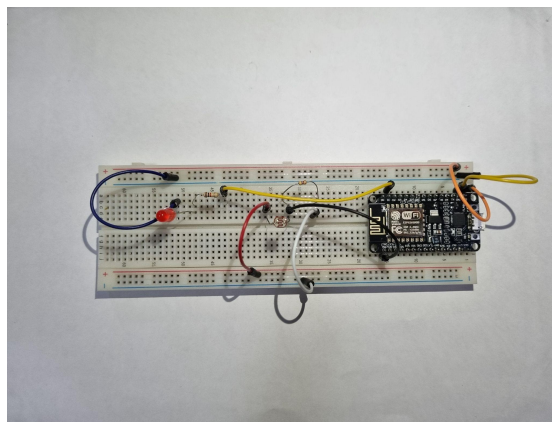
Serial.println("Response from server: " + response);
}
```


Esquema de Conexionado

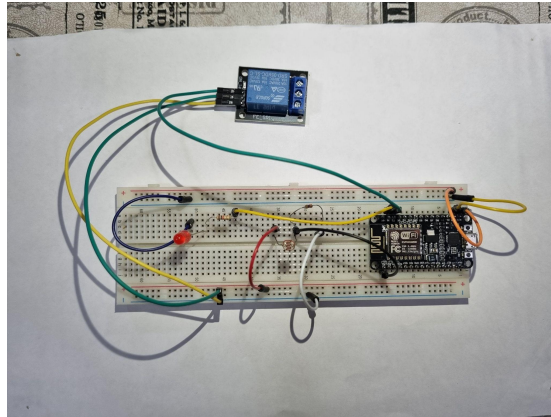
Montamos el módulo ESP8266MOD en la Protoboard y realizamos la conexión de tierra y 3,3 V. en el lado izquierdo de la placa de pruebas.



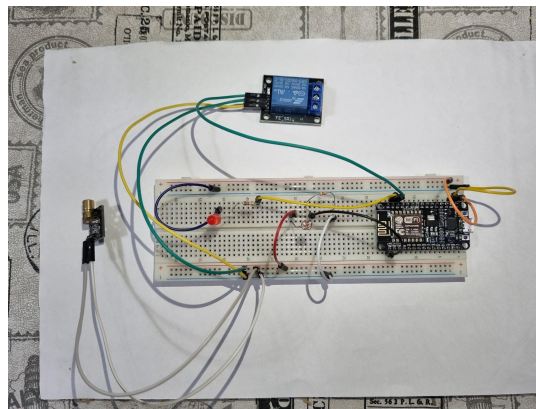
Conectamos el componente LDR, un terminal se conecta al positivo izquierdo, el cual se alimenta con 5V proporcionado por una fuente externa al sistema. El segundo terminal se conecta al pin A0, el cual es el pin analógico con el que realizaremos la toma del valor que indique la LDR; seguido se conecta, dentro de la misma fila del terminal de la LDR una Resistencia hacia GND.



Conectamos un LED, el ánodo (la terminal positiva) se conecta mediante una Resistencia hacia el pin G05, correspondiente al D1 del módulo ESP8266MOD y el cátodo (terminal negativo) se conecta hacia GND (referencia).



Introducimos un Relé, el cual se conecta a la franja de los 5V, tanto el positivo como el negativo, y la señal al pin G04 (D2 correspondiente al módulo ESP), este PIN de señal del módulo Relé funciona como actuador para indicar cuando el LED se enciende tras separar la fuente de iluminación de la LDR.



Por último, se conecta el módulo Láser KY-008, el cual lo vamos a emplear para el alumbrado de la LDR, este se conecta tanto el positivo como negativo hacia la Referencia y 5 V. proporcionados por la fuente externa que alimenta la parte izquierda de la placa de pruebas.



Aca se muestra la fuente externa empleada para alimentar con 5V el lateral izquierdo de la protoboard. A modo ilustrativo se muestra con todas las baterías empleadas, pero en la práctica vamos a utilizar solamente 3 de las mismas para suministrar el total hacia la protoboard.

Conclusiones

Basándonos en los resultados y experiencias que fuimos viviendo en el proceso de desarrollo, determinamos que:

Tiempo de compilación:

El tiempo que se tarda en compilar el código es aproximadamente de 42 segundos.

Conexión WiFi:

La estabilidad de la conexión a WiFi es efectiva, aunque presenta una latencia bastante alta, aproximadamente de 7 segundos aproximadamente, haciendo que los cambios en el sistema se noten luego de varios segundos.

Precisión del LDR y luz laser:

El sensor LDR funciona de manera efectiva siempre y cuando la fuente de energía sea suficiente. Pudimos presenciar que a menor ingreso de energía el LDR registraba valores no deseados. En cuanto al láser tuvimos problemas ya que al comienzo del proyecto el láser tiene una intensidad muy alta pero a medida que pasaron las pruebas su intensidad decreció.

Tiempo que tarda en realizar la acción:

El tiempo que tarda en apagarse completamente el led luego de que se alumbró al LDR es aproximadamente de 9 segundos.

Tiempo que tarda en volver a su estado inicial:

El tiempo que tarda el led en prenderse nuevamente, luego de que se haya dejado de apuntar a la LDR con el láser es aproximadamente de 9 segundos.

En conclusión, en el desarrollo del proyecto nos enfrentamos varios desafíos significativos que se abordaron en su mayoría, permitiendo la implementación del sistema propuesto. Sin embargo, en base a los resultados obtenidos se puede observar que distintas áreas se pueden mejorar para obtener mejores resultados, considerando utilizar componentes alternativos.

Respecto a los resultados obtenidos, el tiempo de compilación se sitúa en aproximadamente 42 segundos. Aunque la conexión WiFi demostró estabilidad, la latencia de alrededor 7 segundos podría impactar la respuesta inmediata del sistema. La precisión del sensor LDR es efectiva en condiciones adecuadas de energía, pero se observaron

inconsistencias a bajos niveles de ingreso de la misma. El láser, inicialmente potente, experimentó una disminución de intensidad durante las pruebas. Además, los tiempos de acción y recuperación del sistema, aproximadamente 9 segundos cada uno, podrían beneficiarse de mejoras para lograr una mayor eficacia y respuesta en tiempo real.

En fin, la superación parcial de los problemas iniciales nos ha proporcionado valiosas lecciones y experiencias que nos podrían guiar en la optimización futura del sistema, señalando la importancia de una evaluación continua y la adaptabilidad en el desarrollo de proyectos tecnológicos.