



UNIVERSIDAD NACIONAL DE CÓRDOBA
FACULTAD DE CIENCIAS EXACTAS, FÍSICAS Y NATURALES
CÁTEDRA DE ELECTRÓNICA DIGITAL III

TRABAJO PRÁCTICO INTEGRADOR

**“SISTEMA DE DETECCIÓN DE ANOMALÍAS EN
NIVELES DE MONÓXIDO DE CARBONO”**

Grupo N° 5

Alumnos:
Costamagna, Matias
de la Mata, Nicolas
Quispe, Mateo

Profesor:
Migliore, Emiliano

Comisión N° 3

18/11/2025

TABLA DE CONTENIDO

- 1. PROYECTO**
- 2. DESCRIPCIÓN DEL SEP**
- 3. DESARROLLO DE HARDWARE**
 - CIRCUITO DEL SSEP MICROCONTROLADOR**
 - CIRCUITO DEL SSE DE SEÑALIZACIÓN ÓPTICA**
 - CIRCUITO DEL SSE DE SEÑALIZACIÓN ACÚSTICA**
 - CIRCUITO DEL SSE DE ADQUISICIÓN DE SEÑALES**
 - CIRCUITO DEL SSE DE COMUNICACIÓN DE DATOS**
 - CIRCUITO DEL SSE DE VISUALIZACIÓN DE DATOS**
- 4. DESARROLLO DE FIRMWARE**
 - FIRMWARE DEL SSE DE SEÑALIZACIÓN ÓPTICA**
 - FIRMWARE DEL SSE DE SEÑALIZACIÓN ACÚSTICA**
 - FIRMWARE DEL SSE DE VISUALIZACIÓN DE DATOS**
 - FIRMWARE DEL SSE DE ADQUISICIÓN DE SEÑALES**
 - FIRMWARE DEL SSE DE COMUNICACIÓN DE DATOS**
- 5. PRUEBAS DE SISTEMA**
- 6. CONCLUSIONES**
- 7. BIBLIOGRAFÍA Y REFERENCIAS**
- 8. ANEXO**
 - CIRCUITO ESQUEMÁTICO DEL SEP**
 - FIRMWARE DEL SEP LPC1769**
 - FIRMWARE DEL SEP ESP8266**
 - LISTA DE MATERIALES**
 - HOJAS DE DATOS**

TÉRMINOS Y DEFINICIONES

TPI: Trabajo Práctico Integrador
SEP: Sistema Electrónico Programable
SSE: Subsistema Electrónico
SSEP: Subsistema Electrónico Programable

1. PROYECTO

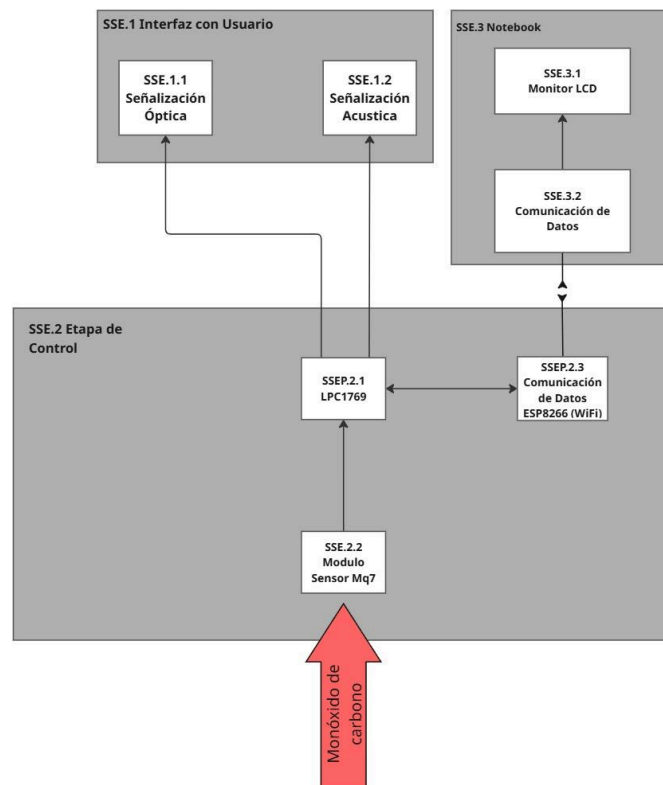
El presente Trabajo Práctico Integrador describe la implementación de un "SISTEMA DE DETECCIÓN DE ANOMALÍAS EN NIVELES DE MONÓXIDO DE CARBONO" destinado a la detección temprana y precisa de concentraciones peligrosas de monóxido de carbono (CO) en ambientes domésticos e industriales.

El sistema se basa en un LPC1769 como microcontrolador principal, el cual adquiere señales analógicas por medio de un sensor de gas MQ-7 especializado en la detección de monóxido de carbono para luego ser procesadas digitalmente. El resultado del análisis permite determinar la concentración de CO en el ambiente, expresada en PPM (partes por millón).

El sistema cuenta con múltiples interfaces de visualización y alerta: un sistema de señalización óptica mediante LEDs de colores que indican visualmente el nivel de peligro (verde: seguro, amarillo: precaución, rojo: peligro crítico), un sistema de señalización acústica mediante buzzer que genera alertas sonoras intermitentes en condiciones críticas, y una interfaz web accesible mediante WiFi a través del módulo ESP8266 que presenta los datos de concentración de CO en tiempo real.

La detección de monóxido de carbono se realiza mediante un sensor MQ-7, el cual posee alta sensibilidad al CO y baja sensibilidad a otros gases.

El sistema implementa además un mecanismo de confirmación de alarma crítica mediante la acumulación de múltiples lecturas consecutivas por encima del umbral, evitando falsas alarmas por picos momentáneos de concentración.



2.DESCRIPCIÓN DEL SEP

El sistema desarrollado consiste en un detector de monóxido de carbono automático basado en el microcontrolador LPC1769, diseñado para monitorear continuamente la concentración de CO en el ambiente mediante un sensor MQ-7. Cuando se detectan niveles peligrosos de gas, el sistema activa alarmas visuales y sonoras de forma progresiva según la gravedad de la situación.

El sistema opera de manera autónoma realizando mediciones periódicas cada 1 [s]. Las lecturas del sensor se procesan mediante el módulo ADC interno del LPC1769, convirtiendo las señales analógicas en valores digitales que representan la concentración de CO en [mV]. Estos valores se utilizan para determinar el estado de alerta del sistema según umbrales configurables: estado seguro (LED verde), estado de precaución (LED amarillo) y estado crítico (LED rojo con alarma sonora).

El usuario recibe información del sistema a través de múltiples interfaces: señalización óptica mediante tres LEDs de colores que indican el nivel de peligro, señalización acústica mediante un buzzer que se activa en situaciones críticas generando un tono de alerta, y transmisión de datos por comunicación serial UART hacia un módulo ESP8266 para visualización remota. Las señales adquiridas del sensor MQ-7 son procesadas por el ADC y distribuidas a los distintos subsistemas: control de alarmas, señalización óptica, señalización acústica y comunicación de datos.

El sistema implementa lógica de histéresis para evitar falsas alarmas, requiriendo múltiples lecturas consecutivas por encima del umbral de precaución antes de activar el estado crítico. Esta característica garantiza mayor confiabilidad en la detección, reduciendo activaciones espurias por variaciones momentáneas en la concentración de gas.

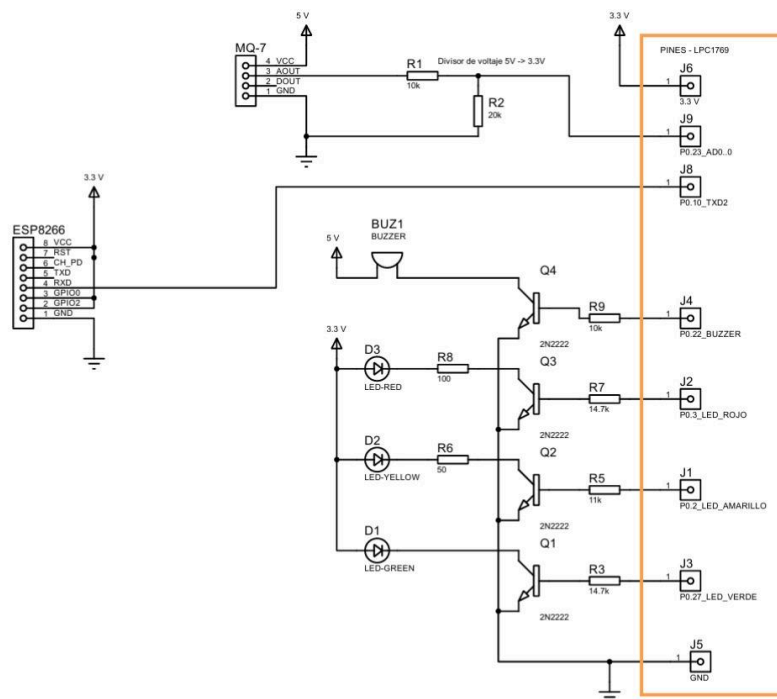
3. DESARROLLO DE HARDWARE

3.1. CIRCUITO DEL SSEP MICROCONTROLADOR

El subsistema electrónico programable central se basa en el microcontrolador LPC1769 de NXP, un dispositivo de 32 bits con arquitectura ARM Cortex-M3 que opera a una frecuencia de 100 [MHz]. Para el clock el sistema utiliza el oscilador RC interno del LPC1769 para generar la señal de reloj del sistema.

El microcontrolador utiliza los siguientes pines para interactuar con los distintos subsistemas:

- P0.23 (AD0.0): Entrada analógica para la lectura del sensor MQ-7, configurada con función alterna 1 para el canal 0 del ADC.
- P1.29 (MAT0.1): Salida del canal de Match 0 del Timer 0, configurada para generar los pulsos de disparo del ADC.
- P0.10 (TXD2): Transmisión UART2, para la comunicación serial con el módulo ESP8266.
- P0.2, P0.3, P0.27: Salidas GPIO configuradas para control de LEDs (rojo, amarillo y verde respectivamente).
- P0.22: Salida GPIO configurada para control del buzzer.



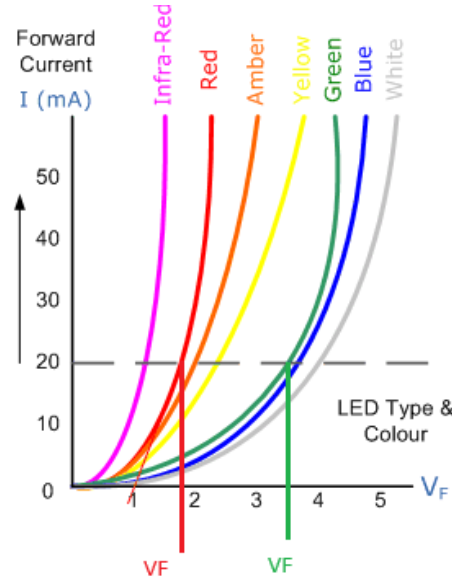
3.2. CIRCUITO DEL SSE DE SEÑALIZACIÓN ÓPTICA

El subsistema de señalización óptica proporciona retroalimentación visual inmediata al usuario sobre el estado de la concentración de monóxido de carbono detectada en el ambiente. Este subsistema utiliza tres LEDs de colores estándar (verde, amarillo y rojo) que indican de manera visual e intuitiva los diferentes niveles de alerta del sistema.

Componentes Utilizados:

- LED Verde D1 (P0.27): Indica estado seguro, con concentración de CO por debajo del umbral de precaución.
- LED Amarillo D2 (P0.3): Indica estado de precaución, con concentración de CO mayor al umbral de precaución.
- LED Rojo D3 (P0.2): Indica estado crítico, con concentración de CO por encima del umbral de precaución sostenida en el tiempo.

Dado que los pines GPIO del LPC1769 tienen una capacidad limitada de corriente (máximo 2.58 [mA] para el periférico GPIO), y considerando que los LEDs estándar requieren típicamente 15-20 [mA] para una luminosidad adecuada, se implementó un circuito mediante transistores NPN 2N2222A para cada LED.



$$R_{LED R} = \frac{V_{CC} - V_{LED R} - V_{CE(sat)}}{I_{LED}} = \frac{3.3 [V] - 1.5 [V] - 0.3 [V]}{15 [mA]} = 93.33 [\Omega] \approx 100 [\Omega]$$

$$R_{LED V} = \frac{V_{CC} - V_{LED V} - V_{CE(sat)}}{I_{LED}} = \frac{3.3 [V] - 3 [V] - 0.3 [V]}{15 [mA]} = 0 [\Omega] \rightarrow \text{No es necesaria}$$

$$R_{LED A} = \frac{V_{CC} - V_{LED A} - V_{CE(sat)}}{I_{LED}} = \frac{3.3 [V] - 2 [V] - 0.3 [V]}{15 [mA]} = 66.67 [\Omega] \approx 50 [\Omega]$$

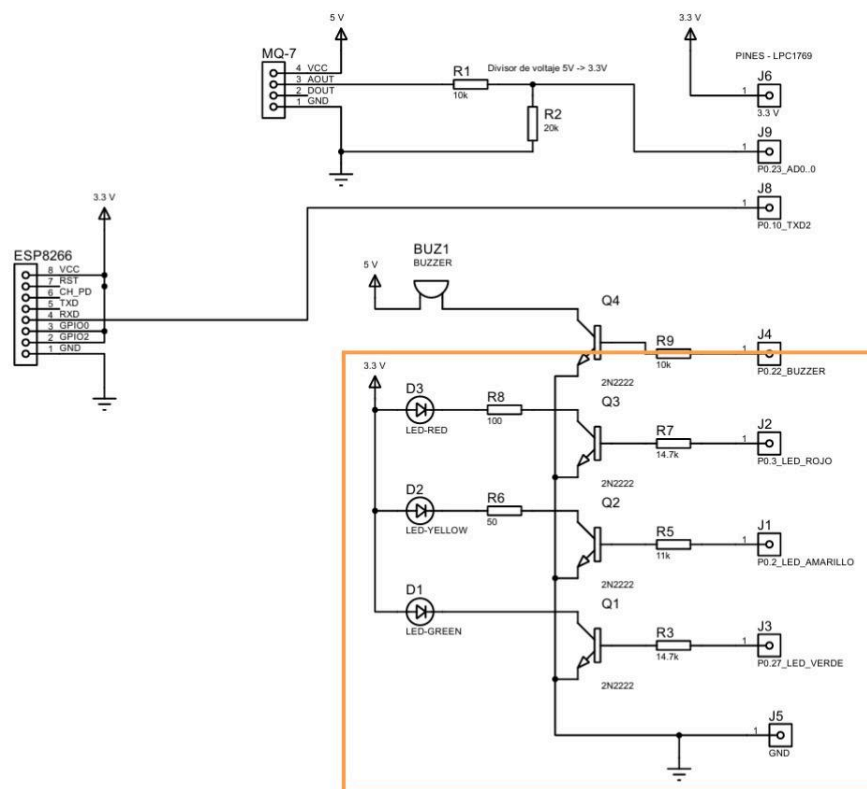
$$I_{LED A} = \frac{V_{CC} - V_{LED A} - V_{CE(sat)}}{R_{LED A}} = \frac{3.3 [V] - 2 [V] - 0.3 [V]}{50 [\Omega]} = 20 [mA]$$

$$R_{base} = \frac{V_{OH} - V_{BE}}{I_B} = \frac{(V_{DD} - 0.4 [V]) - V_{BE}}{I_C / \beta} = \frac{(3.3 [V] - 0.4 [V]) - 0.7 [V]}{15 [mA] / 100} = 14.67 [k\Omega] \approx 14.7 [k\Omega]$$

$$R_{base LED A} = \frac{V_{OH} - V_{BE}}{I_B} = \frac{(V_{DD} - 0.4 [V]) - V_{BE}}{I_C / \beta} = \frac{(3.3 [V] - 0.4 [V]) - 0.7 [V]}{20 [mA] / 100} = 11 [k\Omega]$$

1. Alimentación de los LEDs: Los ánodos de todos los LEDs se conectan directamente a la línea de alimentación de 3.3 [V].
2. Resistencias limitadoras de corriente (R4, R6, R8): Los LEDs rojo y amarillo cuentan con una resistencia en serie de 100 [Ω] y 50 [Ω] respectivamente

- conectadas entre el cátodo de cada LED y el colector del transistor 2N2222A, . Estas resistencias limitan la corriente que circula por los LEDs.
3. Transistores (Q1, Q2, Q3): Se utilizan transistores NPN 2N2222A configurados como interruptores en configuración de emisor común. El colector de cada transistor se conecta al cátodo del LED correspondiente a través de la resistencia limitadora, mientras que el emisor se conecta directamente a tierra (GND).
 4. Resistencias de base (R3, R5, R7): Cada transistor cuenta con una resistencia de base de 14.7 [k Ω] (11 [k Ω] para el LED amarillo) conectada entre el pin GPIO del microcontrolador y la base del transistor. Estas resistencias limitan la corriente de base y protegen el pin GPIO del microcontrolador.



3.3. CIRCUITO DEL SSE DE SEÑALIZACIÓN ACÚSTICA

El subsistema de señalización acústica proporciona una alerta sonora cuando el sistema alcanza el estado crítico. Este subsistema complementa la señalización visual, generando una advertencia audible que permite al usuario ser alertado incluso cuando no está observando directamente los indicadores LED.

Componente Utilizado:

- Buzzer (P0.22): Transductor piezoeléctrico, o buzzer activo, que genera una señal sonora intermitente durante el estado de alarma crítica.

Circuito de Interfaz:

Similar al subsistema de señalización óptica, el buzzer requiere una tensión de alimentación que excede la tensión máxima de suministro del LPC1769 (3.3 [V]). Por esta razón, se implementó un circuito el cual se habilita mediante un transistor NPN 2N2222A que actúa como interruptor.

Datos:

- $V_{CC} = 5$ [V]
- $V_{BUZZER} = 5$ [V] (caída de voltaje típica del buzzer piezoeléctrico)
- $V_{BE} = 0.7$ [V] (voltaje base-emisor del 2N2222A)
- $V_{CE \text{ saturación}} = 0.4$ [V]
- $\beta = 100$
- Corriente de consumo máximo del Buzzer = 30 [mA] = $I_{C \text{ max}}$

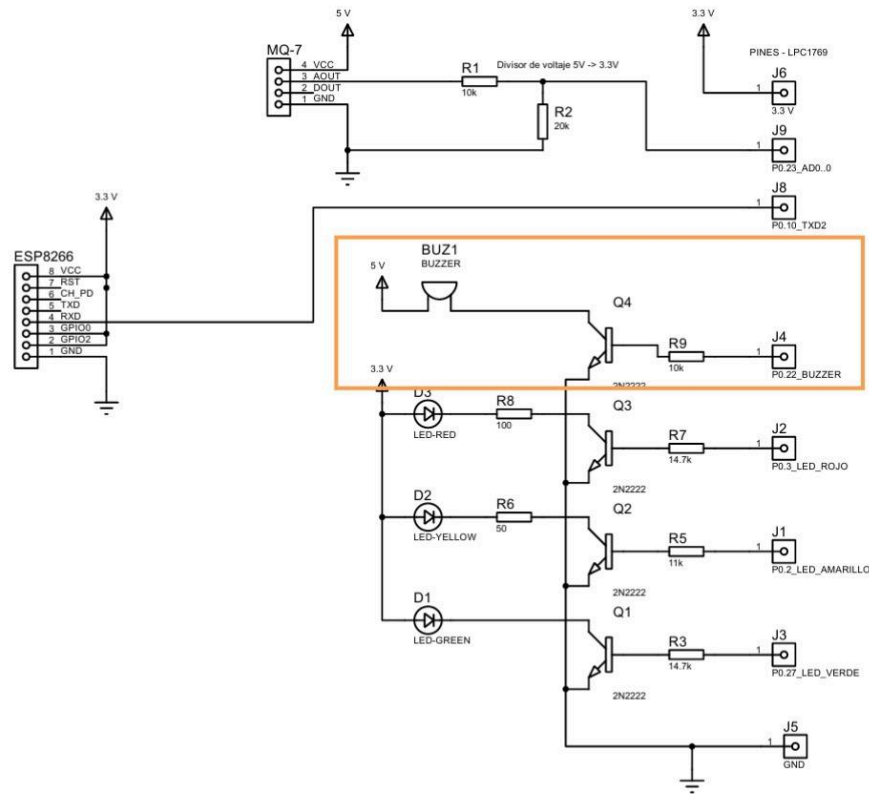
$$I_{B \text{ max}} = \frac{I_{C \text{ max}}}{\beta} = \frac{V_{OH} - V_{BE}}{R_{B \text{ min}}}$$

$$R_{B \text{ min}} = \beta \cdot \frac{V_{OH} - V_{BE}}{I_C} = 100 \cdot \frac{(V_{DD} - 0.4 \text{ [V]}) - 0.7 \text{ [V]}}{30 \text{ [mA]}} = 7.33 \text{ [k}\Omega\text{]} \rightarrow \textit{Se elige } 10 \text{ [k}\Omega\text{]}$$

Debido a que el buzzer genera una caída de tensión de 5V, es necesario alimentarlo con una fuente de tensión externa de ese valor. Además, resulta que no es necesario colocar una resistencia en el colector del transistor, pues la caída de tensión entre el colector y el emisor es menor que la V_{CE} de saturación, lo que nos mantiene en esa zona de operación del transistor.

De esta forma, el buzzer se encenderá cada vez que la salida del pin conectado en la base se encuentre en un nivel lógico "1" generando así la caída de tensión necesaria entre la base y el emisor que habilite el paso de la corriente del colector, funcionando como un switch.

1. Alimentación del buzzer: El terminal positivo del buzzer se conecta directamente a la línea de alimentación de 5 [V].
2. Transistor (Q4): Transistor NPN 2N2222A configurado como interruptor. El colector se conecta al terminal negativo del buzzer, mientras que el emisor se conecta directamente a tierra (GND).
3. Resistencia de base (R9): Una resistencia de 10 [kΩ] conectada entre el pin P0.22 del microcontrolador y la base del transistor Q4. Esta resistencia limita la corriente de base a 0.22 [mA].



3.4. CIRCUITO DEL SSE DE ADQUISICIÓN DE SEÑALES

El subsistema de adquisición de señales es el componente sensorial del sistema, responsable de detectar la presencia y concentración de monóxido de carbono en el ambiente. Este subsistema utiliza el sensor de gas MQ-7, capaz de detectar concentraciones en el rango de 20 a 2000 [ppm]. El sensor MQ-7 requiere una alimentación de 5 [V] para su correcto funcionamiento. Esta tensión se obtiene directamente de la línea de 5 [V] del puerto USB de la computadora.

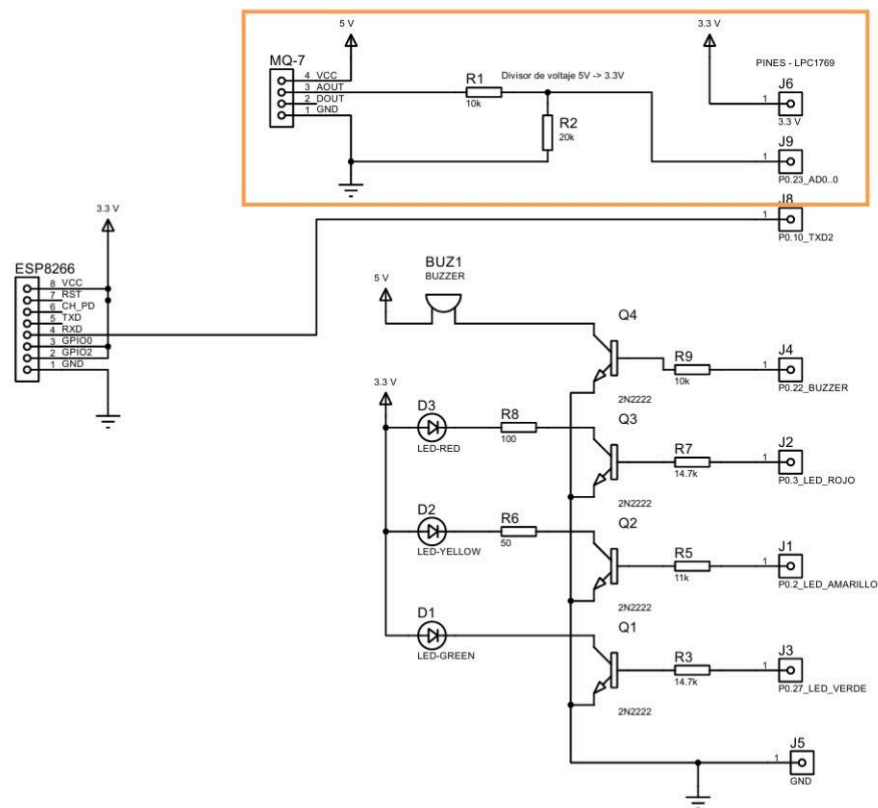
La salida analógica del sensor MQ-7 proporciona un voltaje proporcional a la concentración de CO detectada, con un rango típico de 0 a 5 [V]. Sin embargo, el conversor analógico-digital (ADC) del microcontrolador LPC1769 acepta señales de entrada con un rango dinámico comprendido entre 0 y 3.3 [V]. Aplicar voltajes superiores a 3.3 [V] en el pin del ADC podría dañar permanentemente el microcontrolador.

Para resolver esta incompatibilidad, se implementó un circuito divisor resistivo que escala proporcionalmente el voltaje de salida del sensor al rango aceptable por el ADC. El divisor está compuesto por dos resistencias:

$$V_{OUT} = V_{IN} \cdot \frac{R_2}{R_1 + R_2} = 5 [V] \cdot \frac{20[k\Omega]}{10[k\Omega] + 20[k\Omega]} = 3.3[V]$$

- R1 = 10 [kΩ]: Resistencia conectada entre la salida del sensor y el pin P0.23 (AD0.0) del microcontrolador.
- R2 = 20 [kΩ]: Resistencia conectada entre el pin P0.23 (AD0.0) y tierra (GND).

La señal acondicionada se conecta al pin P0.23 del microcontrolador LPC1769, configurado como entrada analógica (AD0.0), correspondiente al canal 0 del convertor analógico-digital.



Calibración del Sensor MQ-7

La calibración del sensor MQ-7 se realizó utilizando la curva característica proporcionada en la hoja de datos del fabricante, que relaciona la razón (R_s/R_o) con la concentración de monóxido de carbono en partes por millón (PPM).

R_s es el valor de la resistencia en presencia de la concentración actual de la sustancia mientras que R_o es el valor de resistencia del sensor en presencia de una concentración de 100 [ppm] de CO.

A partir de los puntos de la curva característica, se construyó una tabla de valores PPM en función de la relación R_s/R_o y se graficaron los datos para identificar el comportamiento del sensor. Mediante ajuste de curvas, se determinó que la relación entre la razón R_s/R_o y la concentración de CO es la siguiente:

$$\text{Concentración [PPM]} = 100 \cdot \left(\frac{R_s}{R_o}\right)^{-1.52}$$

Esta ecuación permite convertir directamente las lecturas de resistencia del sensor en valores de concentración de CO. El valor de R_o lo obtuvimos midiendo en una zona sin presencia de CO agregado, fue de 72.41 [kΩ].

El valor de tensión entregado por el circuito de acondicionamiento es el valor de la caída de tensión producida sobre una resistencia de carga R_L , que en nuestro caso es de 47 [kΩ].

En el firmware, la resistencia del sensor R_s se calcula a partir del voltaje que ingresa por el pin del ADC.

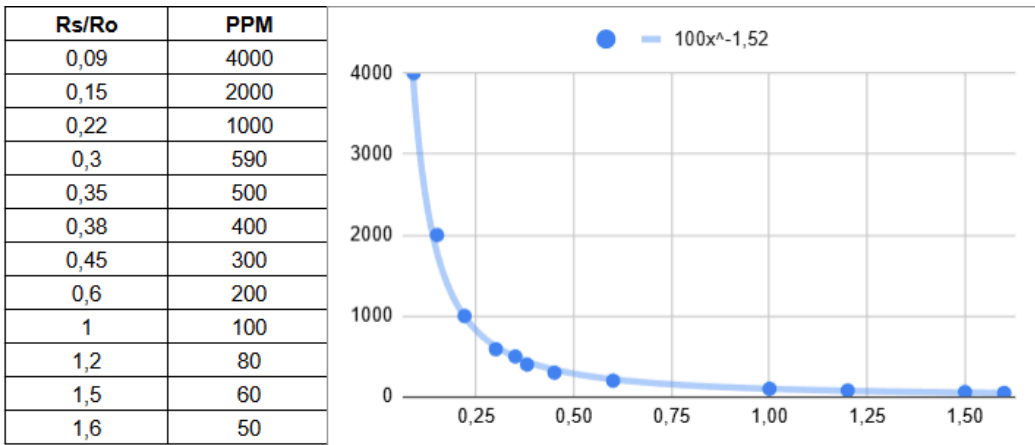
La fórmula que permite calcular la resistencia viene dada por la siguiente expresión:

$$\frac{R_s}{R_L} = \frac{(V_c - V_{RL})}{V_{RL}}$$

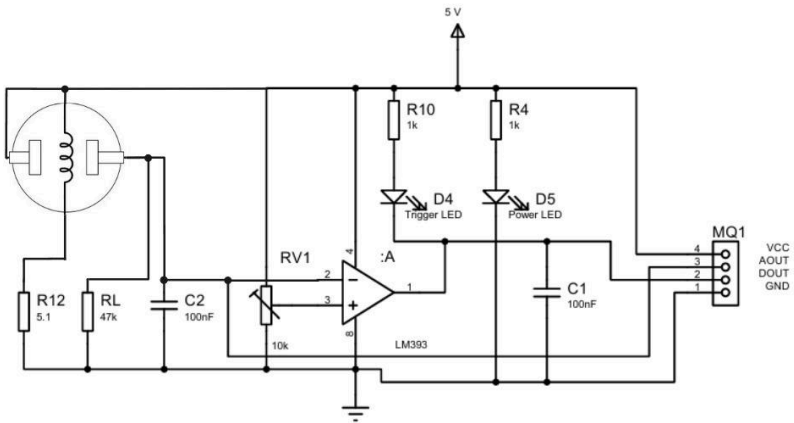
$$R_s = R_L \cdot \frac{(V_c - V_{RL})}{V_{RL}}$$

Una vez estimado el valor de R_s , se obtiene la relación con respecto a R_o y ese resultado se ingresa en la fórmula que calcula la concentración a partir de la relación entre R_s y R_o .

La implementación de esta conversión se encuentra en la función ***convert_adc_to_ppm()*** y es la que permite que el sistema determine en tiempo real la concentración de CO en el ambiente con precisión suficiente para activar las alarmas según los umbrales configurados.



Esquemático del módulo MQ-7:



3.5. CIRCUITO DEL SSE DE COMUNICACIÓN DE DATOS

El subsistema de comunicación de datos desde la placa a la PC se realiza en 2 tramos: El LPC1769 envía la información al módulo ESP8266 utilizando comunicación serial UART; el ESP8266 funciona como puente de red, reenviando los respectivos datos hacia la PC del usuario a través de una conexión WiFi.

Para implementar esta comunicación, el sistema utiliza el periférico UART2 del microcontrolador. La UART es un protocolo de comunicación serial asíncrono que no requiere señal de reloj compartida entre transmisor y receptor, simplificando las conexiones de hardware.

El sistema implementa un esquema de transmisión alternada controlado por el Timer 2, que gestiona dos modos de operación:

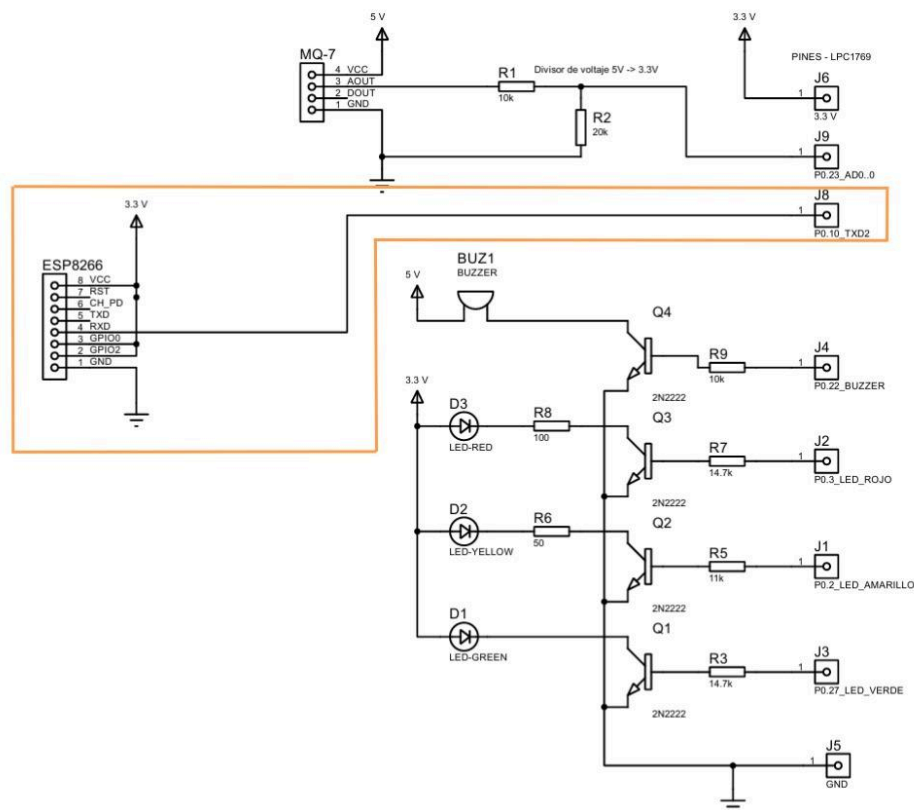
- Modo de transmisión en tiempo real: Transmite datos instantáneos cada 1 [s] durante un período de 10 [s]. En este modo, se envían las lecturas individuales del sensor mediante tramas tipo 'U'.
- Modo de transmisión de promedios: Activo durante 3 [s], en el cual se transmite el promedio de las últimas 10 muestras capturadas mediante tramas tipo 'P'. Durante este período, el sistema utiliza el DMA para transferir las muestras almacenadas a un buffer secundario y a partir de allí calcular el promedio.

El Timer 2 alterna automáticamente entre estos dos modos, proporcionando al usuario tanto información instantánea como datos estadísticos promediados que permiten una evaluación más precisa de la concentración de CO en el ambiente.

Los parámetros de configuración de la UART2 son:

- Velocidad de transmisión (Baud Rate): 9600 bps, velocidad estándar que garantiza comunicación confiable.
- Bits de datos: 8 bits por carácter (configuración estándar).
- Bit de paridad: Ninguno.
- Bits de parada: 1 bit (configuración estándar).
- Control de flujo: Ninguno, la comunicación es unidireccional desde el LPC1769 hacia el ESP8266.

Para la conexión se utiliza únicamente la línea de transmisión (TX) del UART2 por medio del pin P0.10. Este pin se conecta directamente al pin RX (receptor) del módulo ESP8266.



3.6. CIRCUITO DEL SSE DE VISUALIZACIÓN DE DATOS

El subsistema de visualización de datos proporciona al usuario acceso remoto a la información de concentración de monóxido de carbono detectada por el sistema. Se implementa una solución de visualización basada en conectividad WiFi mediante el módulo ESP8266, permitiendo el monitoreo desde cualquier dispositivo con navegador web dentro de la red local. Componentes del Subsistema:

Módulo ESP8266:

- Actúa como servidor web que aloja la interfaz de usuario
- Recibe datos del LPC1769 a través de UART2 a 9600 bps
- Proporciona conectividad WiFi
- Alimentación: 3.3V (compatible con niveles lógicos del LPC1769)

Protocolo de Comunicación:

El LPC1769 envía tramas de datos al ESP8266 mediante comunicación serial con el siguiente formato:

- Trama tipo 'U': Datos instantáneos - U<valor ppm>\n
 - Ejemplo: U15\n indica 15 [ppm] de CO detectado en la última lectura individual
 - Se transmite cada 1 [s] durante los períodos de 10 [s] en modo tiempo real

- Trama tipo 'P': Datos promediados - P<valor_ppm>\n
 - Ejemplo: P12\n indica 12 [ppm] de CO como promedio de 10 muestras
 - Se transmite durante los períodos de 3 [s] en modo promedio
 - El cálculo del promedio se realiza mediante transferencia DMA de las muestras almacenadas

El sistema alterna automáticamente entre el envío de datos instantáneos (10 [s]) y datos promediados (3 [s]), permitiendo al usuario visualizar tanto la evolución inmediata como las tendencias estadísticas de la concentración de CO.

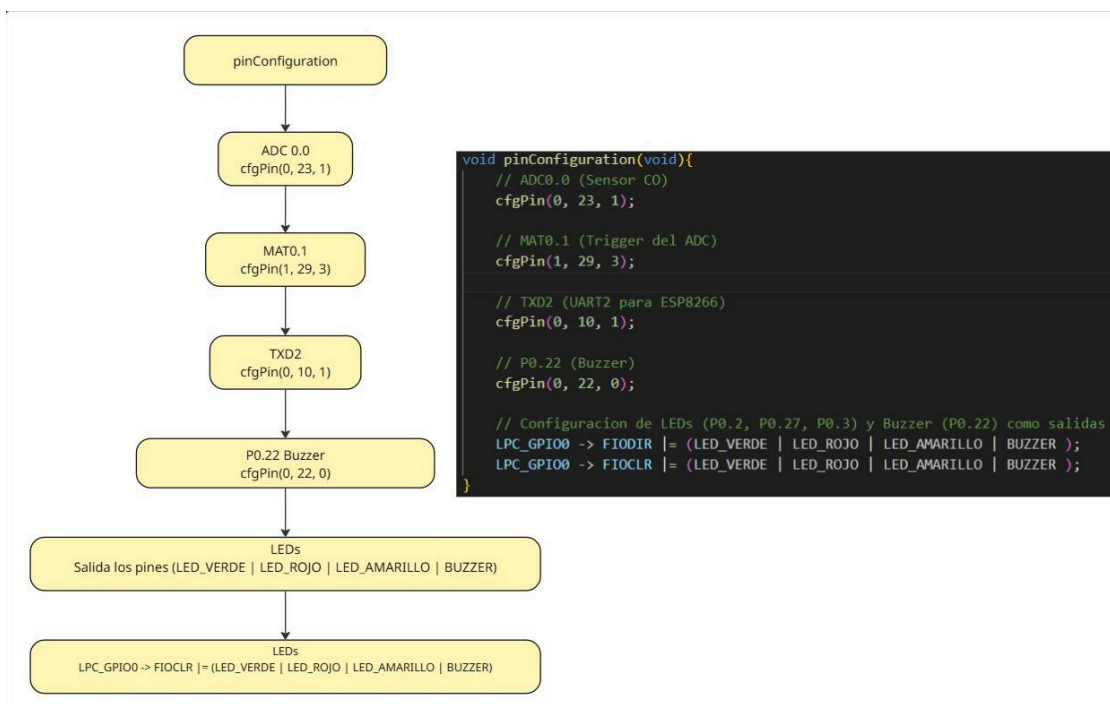
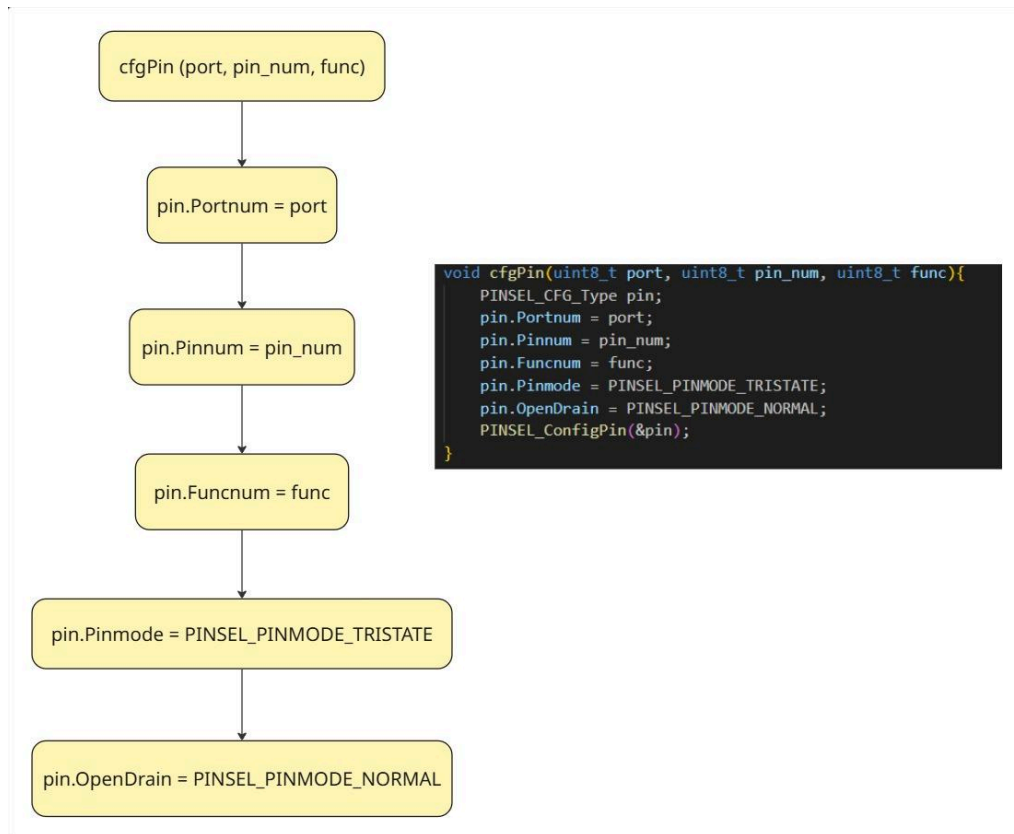
Interfaz Web:

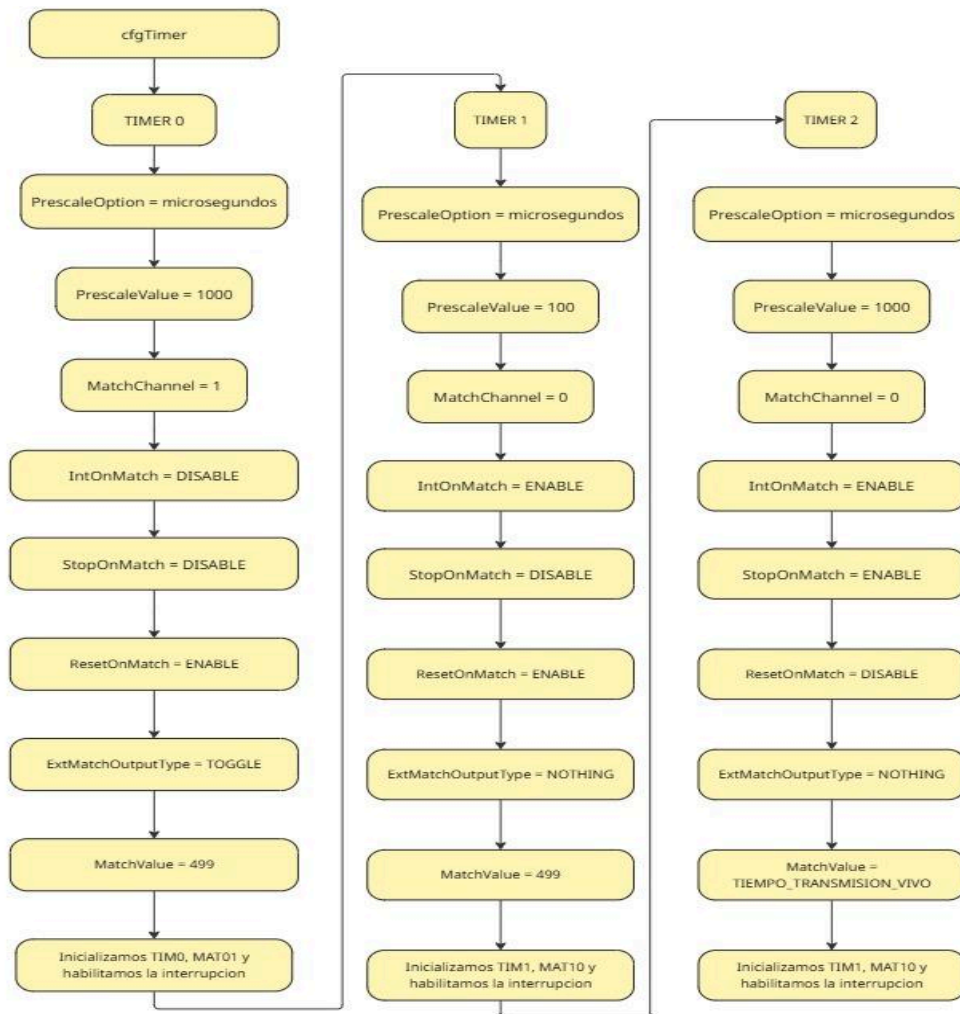
El subsistema de visualización presenta una interfaz web HTML alojada en el ESP8266 que muestra:

- Concentración actual de CO en [ppm]: Valor numérico actualizado en tiempo real
- Gráfico temporal: Representación visual de la evolución de la concentración de CO
- Indicador de estado: Representación visual del nivel de alarma (seguro/precaución/crítico)
- Tipo de medición: Indicación de la última medición instantánea y el último promedio obtenido

4. DESARROLLO DE FIRMWARE

4.1. CONFIGURACIÓN GENERAL DEL FIRMWARE GENERAL (Pines, ADC, UART, Timers)





```

void cfgTimer(void){
    // --- Timer 0: Generador de Pulsos (Trigger ADC y Envio UART) ---
    TIM_TIMERCFG_Type timerMode0;
    timerMode0.PrescaleOption = TIM_PRESCALE_USVAL;
    timerMode0.PrescaleValue = 1000; //Base de tiempo de 1ms

    TIM_MATCHCFG_Type timerMAT01; // Para iniciar ADC y la Interrupcion de Envio UART
    timerMAT01.MatchChannel = 1;
    timerMAT01.IntOnMatch = DISABLE; // Generar interrupcion para el envio UART
    timerMAT01.StopOnMatch = DISABLE;
    timerMAT01.ResetOnMatch = ENABLE;
    timerMAT01.ExtMatchOutputType = TIM_EXTMATCH_TOGGLE; // trigger para el ADC
    timerMAT01.MatchValue = 499; // Interrumpe y dispara ADC cada 500ms (2Hz)

    TIM_Init(LPC_TIM0, TIM_TIMER_MODE, &timerMode0);
    TIM_ConfigMatch(LPC_TIM0, &timerMAT01);
    TIM_Cmd(LPC_TIM0, ENABLE);

    // --- Timer 1: Generador de Tono para Buzzer (Alarma) ---
    // Se usa como base de tiempo para alternar el Buzzer en el ISR de TIMER1
    TIM_TIMERCFG_Type timerModel;
    timerModel.PrescaleOption = TIM_PRESCALE_USVAL;
    timerModel.PrescaleValue = 100; // Base de tiempo de 100us

    TIM_MATCHCFG_Type timerMAT10;
    timerMAT10.MatchChannel = 0;
    timerMAT10.IntOnMatch = ENABLE;
    timerMAT10.StopOnMatch = DISABLE;
    timerMAT10.ResetOnMatch = ENABLE;
    timerMAT10.ExtMatchOutputType = TIM_EXTMATCH_NOTHING;
    timerMAT10.MatchValue = 499; // Genera interrupcion cada 500ms

    TIM_Init(LPC_TIM1, TIM_TIMER_MODE, &timerModel);
    TIM_ConfigMatch(LPC_TIM1, &timerMAT10);
    NVIC_EnableIRQ(TIMER1_IRQn);

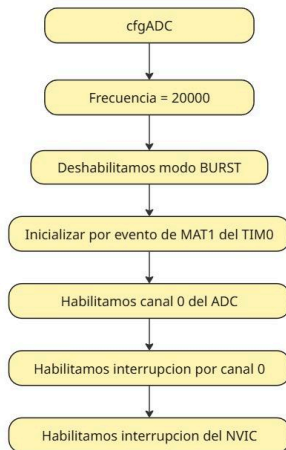
    // --- Timer 2: Cambia el valor a transmitir entre la transmision en tiempo real y el promedio ---
    TIM_TIMERCFG_Type timerMode2;
    timerMode2.PrescaleOption = TIM_PRESCALE_USVAL;
    timerMode2.PrescaleValue = 1000; //Base de tiempo de 1ms

    TIM_MATCHCFG_Type timerMAT20;
    timerMAT20.MatchChannel = 0;
    timerMAT20.IntOnMatch = ENABLE;
    timerMAT20.StopOnMatch = ENABLE;
    timerMAT20.ResetOnMatch = DISABLE;
    timerMAT20.ExtMatchOutputType = TIM_EXTMATCH_NOTHING; // trigger para el ADC
    timerMAT20.MatchValue = 10000; // Tiempo inicial 10 segundos

    TIM_Init(LPC_TIM2, TIM_TIMER_MODE, &timerMode2);
    TIM_ConfigMatch(LPC_TIM2, &timerMAT20);
    TIM_Cmd(LPC_TIM2, ENABLE);

    NVIC_EnableIRQ(TIMER2_IRQn);
}

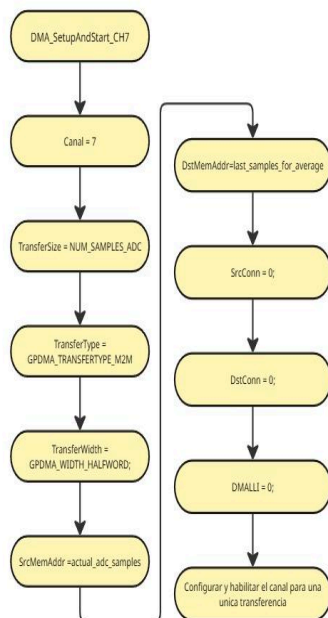
```

```

void cfgADC(void){
    ADC_Init(LPC_ADC, ADC_FREQ);
    ADC_BurstCmd(LPC_ADC, DISABLE); // Deshabilitar modo Burst
    ADC_StartCmd(LPC_ADC, ADC_START_ON_MAT01); // Iniciar por Match1 del Timer 0
    ADC_EdgeStartConfig(LPC_ADC, ADC_START_ON_RISING);
    ADC_ChannelCmd(LPC_ADC, ADC_CHANNEL_0, ENABLE); // Habilitar canal 0
    ADC_IntConfig(LPC_ADC, ADC_ADINTEN0, ENABLE); // Habilitar interrupcion en canal 0

    NVIC_EnableIRQ(ADC_IRQn);
}
  
```

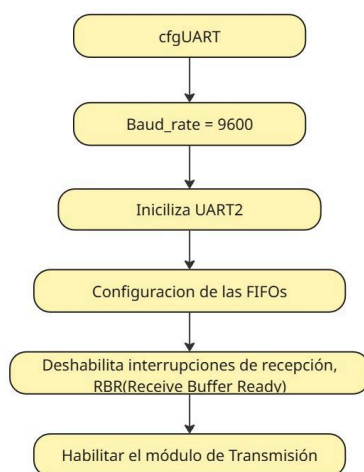


```

void DMA_SetupAndStart_CH7(void){
    // Reconfigura el canal DMA 7 (M2M) para asegurar que el contador de transferencia se reinicie.
    GPDMA_Channel_CFG_Type cfgMEM_DMA_CH7;

    cfgMEM_DMA_CH7.ChannelNum = 7;
    cfgMEM_DMA_CH7.TransferSize = NUM_SAMPLES_ADC;
    cfgMEM_DMA_CH7.TransferType = GPDMA_TRANSFERTYPE_M2M;
    // Usar Half-Word (16 bits) ya que los buffers son uint16_t
    cfgMEM_DMA_CH7.TransferWidth = GPDMA_WIDTH_HALFWORD;
    cfgMEM_DMA_CH7.SrcMemAddr = (uint32_t)actual_adc_samples;
    cfgMEM_DMA_CH7.DstMemAddr = (uint32_t)last_samples_for_average;
    cfgMEM_DMA_CH7.SrcConn = 0;
    cfgMEM_DMA_CH7.DstConn = 0;
    cfgMEM_DMA_CH7.DMALLI = 0;

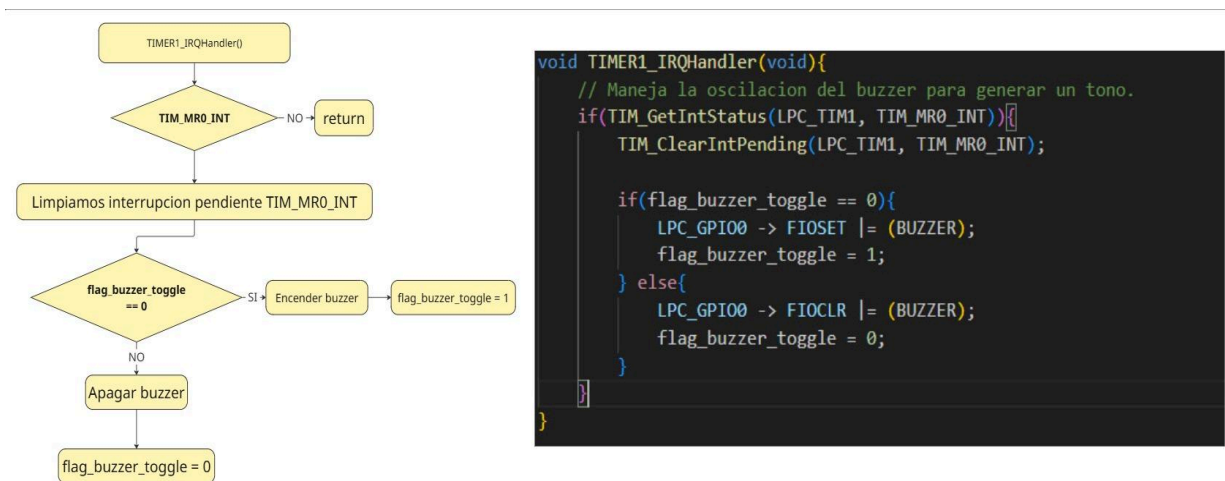
    // Configurar y habilitar el canal para una unica transferencia
    GPDMA_Setup(&cfgMEM_DMA_CH7);
    GPDMA_ChannelCmd(7, ENABLE);
    NVIC_EnableIRQ(DMA_IRQn);
}
  
```

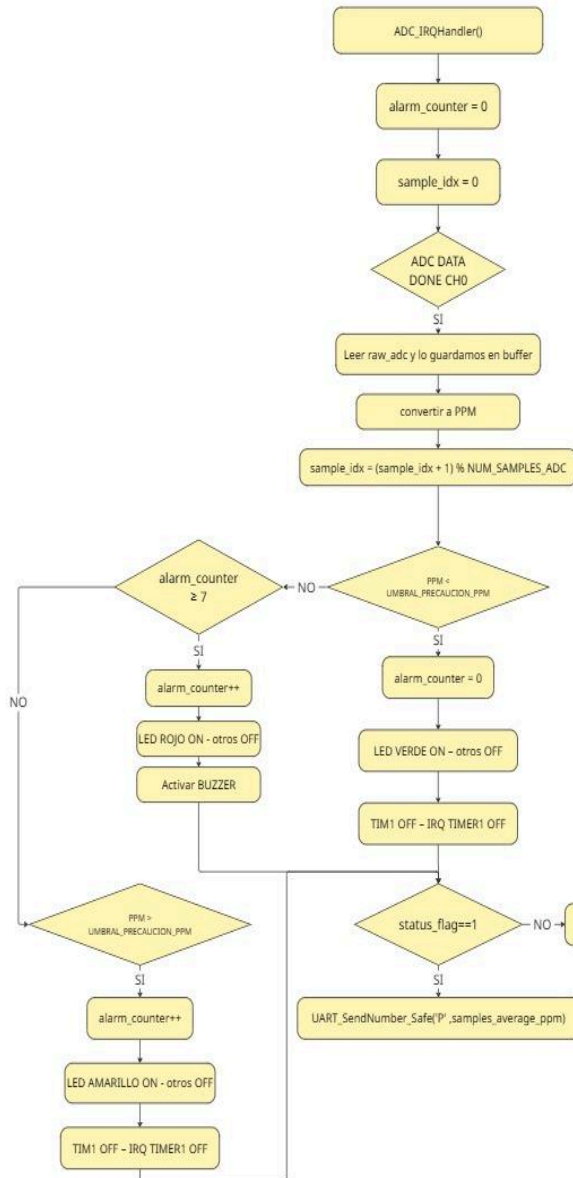


```

void cfgUART(void){
    UART_CFG_Type pinUART2;
    UART_FIFO_CFG_Type UARTFIFO;
    pinUART2.Baud_rate = 9600;
    UART_ConfigStructInit(&pinUART2);
    UART_Init(LPC_UART2, &pinUART2);
    UART_FIFOConfigStructInit(&UARTFIFO);
    UART_FIFOConfig(LPC_UART2, &UARTFIFO);
    UART_IntConfig(LPC_UART2, UART_INTCFG_RBR, DISABLE);
    UART_TxCmd(LPC_UART2, ENABLE);
}
  
```

4.2. FIRMWARE DEL SSE DE SEÑALIZACIÓN ÓPTICA Y ACÚSTICA





```

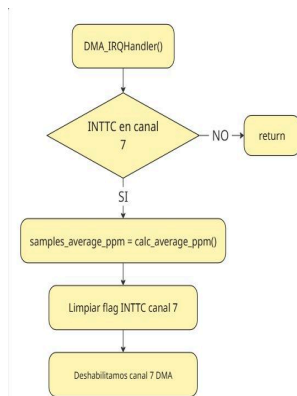
void ADC_IRQHandler(void){
    // Verificamos si la interrupción por canal 0 se disparó.
    if(ADC_ChannelGetStatus(LPC_ADC, ADC_CHANNEL_0, ADC_DATA_DONE)){
        // 1. CAPTURA Y ALMACENAMIENTO DE DATO
        // Leemos el dato, quitamos los 4 bits de status y lo guardamos en el buffer de 16 bits.
        actual_adc_samples[sample_idx] = (ADC_ChannelGetData(LPC_ADC, ADC_CHANNEL_0)) >> 4;
        // 2. CONVERSION Y LÓGICA DE TIEMPO REAL
        last_adc_value_ppm = convert_adc_to_ppm(actual_adc_samples[sample_idx]);

        // 3. ACTUALIZAR INDICE (Captura circular manual)
        sample_idx = (sample_idx + 1) % NUM_SAMPLES_ADC;

        // --- Logica de Alarmas ---
        if(last_adc_value_ppm < UMBRAL_PRECAUCION_PPM){
            // Estado SEGURO: LED Verde
            alarm_counter = 0;
            LPC_GPIO0 -> FIOSET |= (LED_VERDE);
            LPC_GPIO0 -> FIOCLR |= (LED_AMARILLO | LED_ROJO | BUZZER);
            TIM_Cmd(LPC_TIM1, DISABLE);
            NVIC_DisableIRQ(TIM1_IRQn);
        }
        else if(alarm_counter >= 7){
            alarm_counter++;
            // Estado CRITICO: LED Rojo + Buzzer
            LPC_GPIO0 -> FIOCLR |= (LED_AMARILLO);
            LPC_GPIO0 -> FIOSET |= (LED_ROJO);
            TIM_Cmd(LPC_TIM1, ENABLE); // Activar buzzer con PWM
            NVIC_EnableIRQ(TIM1_IRQn);
        }
        else if(last_adc_value_ppm >= UMBRAL_PRECAUCION_PPM){
            // Estado PRECAUCION: LED Amarillo
            alarm_counter++; // Acumular lecturas de precaucion
            LPC_GPIO0 -> FIOCLR |= (LED_ROJO | BUZZER);
            LPC_GPIO0 -> FIOSET |= (LED_AMARILLO);
            TIM_Cmd(LPC_TIM1, DISABLE);
            NVIC_DisableIRQ(TIM1_IRQn);
        }
    }

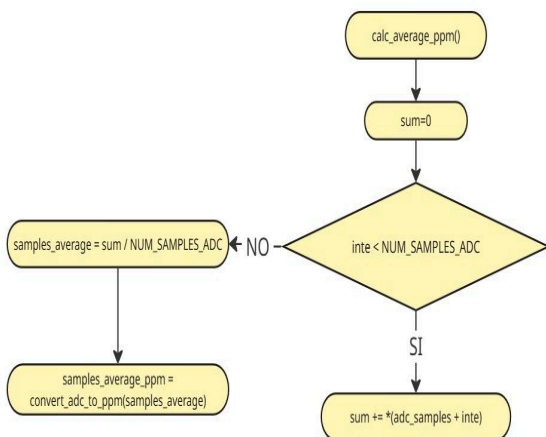
    // USO DE LA FUNCION UART SEGURA (HardFault Fix)
    if(status_flag){
        UART_SendNumber_Safe('P', samples_average_ppm);
    }
    else{
        UART_SendNumber_Safe('U', last_adc_value_ppm);
    }
}
  
```

4.3. FIRMWARE DEL SSE DE ADQUISICIÓN DE SEÑALES



```

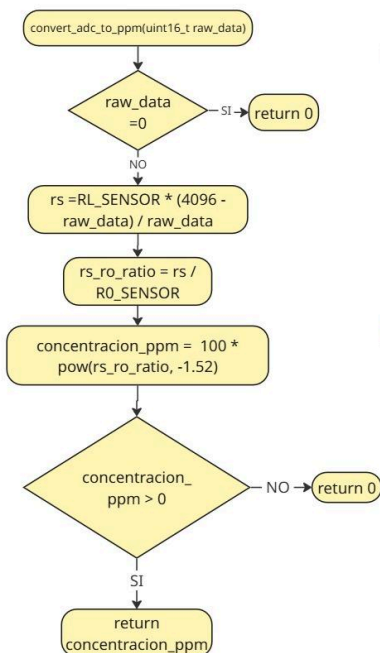
void DMA_IRQHandler(void){
    if(GPDMA_IntGetStatus(GPDMA_STAT_INTTC, 7)){
        samples_average_ppm = calc_average_ppm();
        GPDMA_ClearIntPending(GPDMA_STAT_INTTC, 7);
        GPDMA_ChannelCmd(7, DISABLE);
    }
}
  
```



```

uint16_t calc_average_ppm(void){
    uint32_t sum = 0;

    // Iteracion sobre el banco de memoria
    for(uint16_t inte = 0; inte < NUM_SAMPLES_ADC; inte++){
        // Extraccion del valor del ADC de 12 bits de la palabra de 32 bits (bits 4-15)
        sum += (*(last_samples + inte) >> 4) & 0x0FFF;
    }
    uint16_t samples_average = (uint16_t) sum / NUM_SAMPLES_ADC;
    return convert_adc_to_ppm(samples_average);
}
  
```



```

uint16_t convert_adc_to_ppm(uint16_t raw_data){
    // El valor raw es de 12 bits (0-4095)
    if (raw_data == 0) return 0;

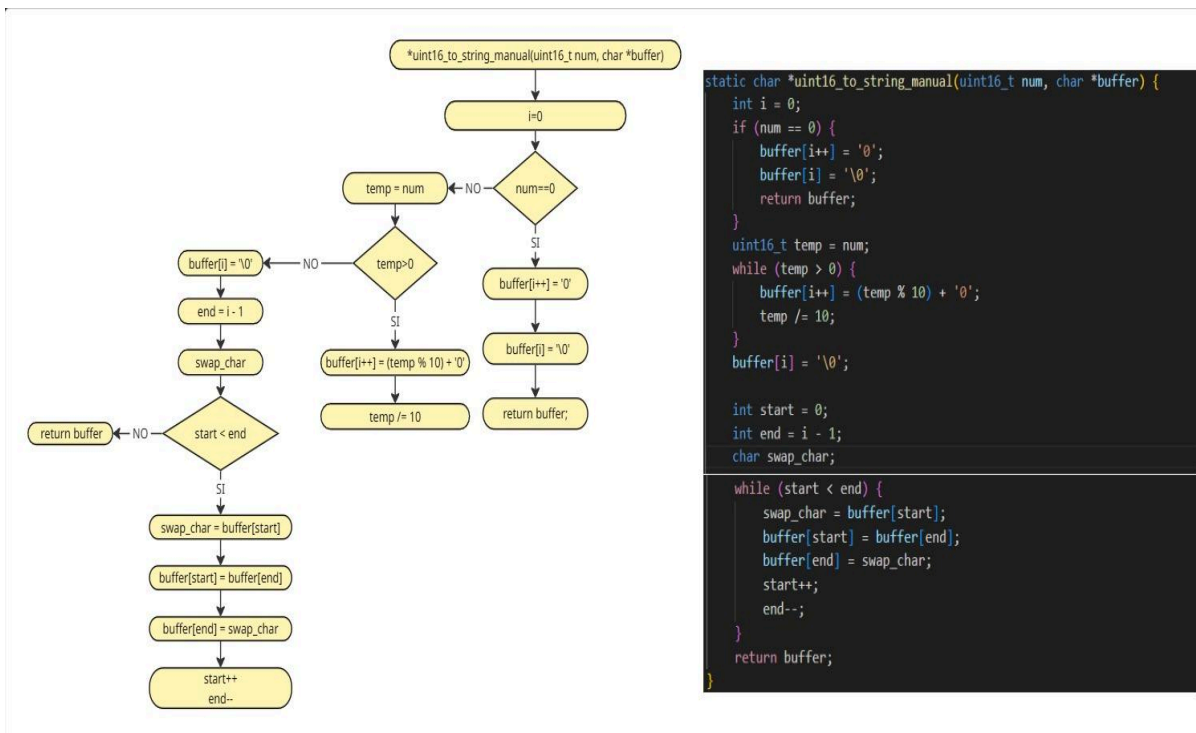
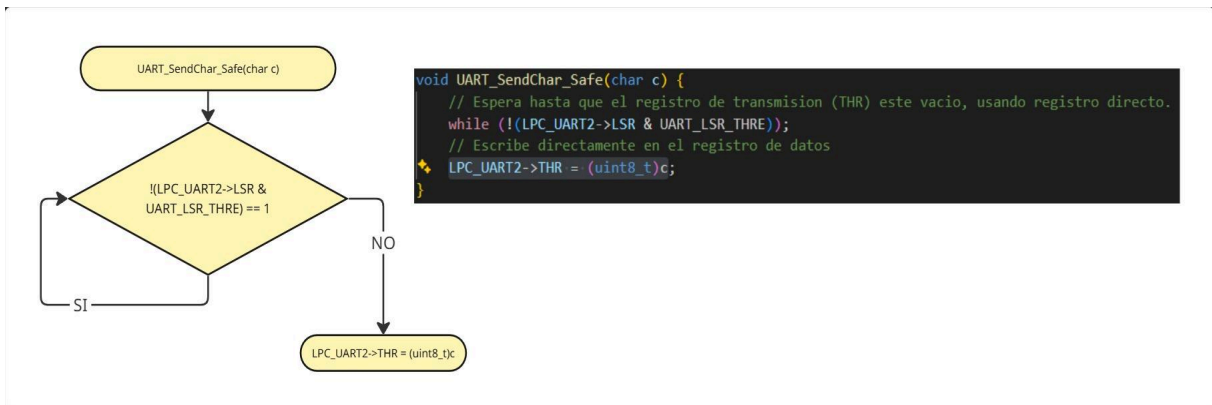
    // Voltaje de salida del sensor Vs = (raw_data / 4096) * 3.3V
    // Resistencia del sensor Rs = (3.3V - Vs) / (Vs / RL_SENSOR)
    // Resistencia del sensor Rs = RL_SENSOR * (4096 - raw_data) / raw_data
    float rs = (float) RL_SENSOR * (4096 - raw_data) / raw_data;

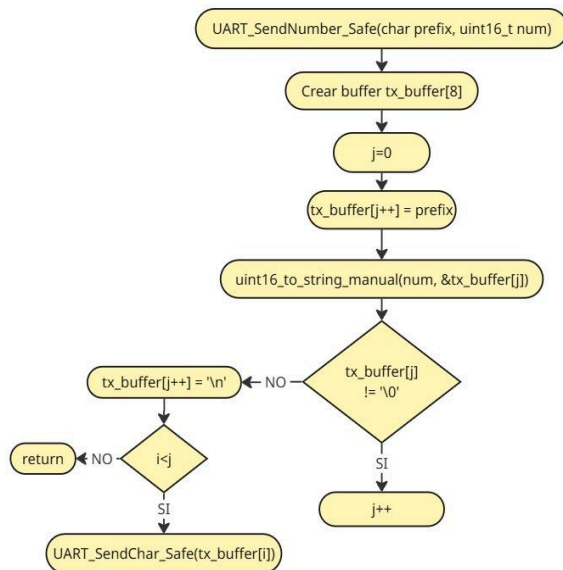
    // Relacion Rs/R0
    float rs_ro_ratio = rs / R0_SENSOR;

    uint16_t concentracion_ppm = (uint16_t) 100.0f * pow(rs_ro_ratio, -1.52f);

    // Asegurar un valor positivo
    return (concentracion_ppm > 0) ? concentracion_ppm : 0;
}
  
```

4.4. FIRMWARE DEL SSE DE COMUNICACIÓN DE DATOS





```

void UART_SendNumber_Safe(char prefix, uint16_t num){
    char tx_buffer[8];
    int j = 0;

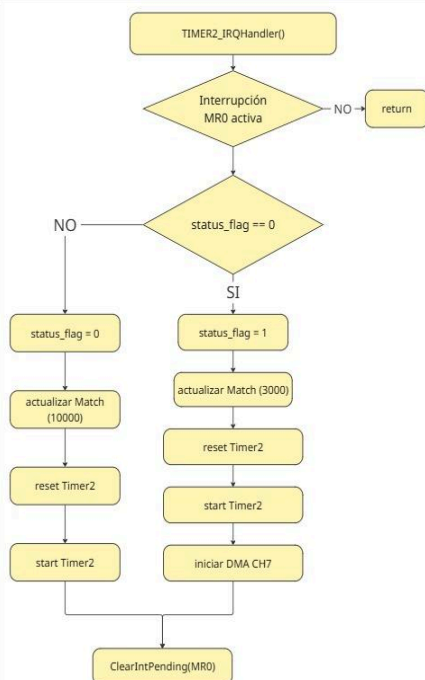
    tx_buffer[j++] = prefix;

    uint16_to_string_manual(num, &tx_buffer[j]);

    while(tx_buffer[j] != '\0') {
        j++;
    }

    tx_buffer[j++] = '\n';

    for (int i = 0; i < j; i++) {
        UART_SendChar_Safe(tx_buffer[i]);
    }
}
  
```



```

void TIMER2_IRQHandler(void){
    if(TIM_GetIntStatus(LPC_TIM2, TIM_MR0_INT)){
        if(status_flag == 0){
            status_flag = 1;
            TIM_UpdateMatchValue(LPC_TIM2, 0, 3000);
            TIM_ResetCounter(LPC_TIM2);
            TIM_Cmd(LPC_TIM2, ENABLE);

            DMA_SetupAndStart_CH7(); // Reinicia la transferencia de los datos
        } else{
            status_flag = 0;
            TIM_UpdateMatchValue(LPC_TIM2, 0, 10000);
            TIM_ResetCounter(LPC_TIM2);
            TIM_Cmd(LPC_TIM2, ENABLE);
        }
        TIM_ClearIntPending(LPC_TIM2, TIM_MR0_INT);
    }
}
  
```


5. PRUEBAS

Nro. Caso de Prueba	Descripción del Caso de Prueba	Paso	Resultado Esperado	Resultado Obtenido	Observaciones
1	Pruebas de Señalización Visual (LEDs)	Mantener sensor en aire limpio (<25 [ppm])	LED verde encendido, LED amarillo y rojo apagados	PASO	Valor estable de concentración obtenido 2 [ppm]
2	Pruebas de Señalización Acústica (Buzzer)	Simular concentración crítica (> 25 [ppm] por 10 lecturas)	Buzzer genera tono intermitente (~2 [Hz])	PASO	Distinción notable en la intermitencia del buzzer
3	Pruebas de Comunicación UART	Capturar datos UART en monitor serial a 9600 [bps]	Distinción de cada tipo de trama ("U" o "P")	PASO	Se distingue correctamente el tipo de trama en su correspondiente periodo
4	Pruebas de Lógica de Sistema	Generar 6 lecturas > UMBRAL SEGURO seguidas	Sistema en precaución, NO activa alarma crítica	PASO	Se alterna correctamente entre el estado seguro y de precaución, según corresponda la lectura, sin alcanzar el estado de alarma
5	Pruebas de Interfaz Web (ESP8266)	Encender ESP8266 y verificar conexión	Página de resultados visible	PASO	Pequeña demora en la visibilidad de los datos debido a la conexión del módulo WiFi con la red

6. CONCLUSIONES

Este TPI logró implementar exitosamente un sistema funcional de detección de monóxido de carbono, cumpliendo con los objetivos planteados mediante la integración de los módulos principales del LPC1769. Destacamos como aprendizajes significativos la implementación de lógica de histéresis para evitar falsas alarmas y la calibración del sensor analógico.

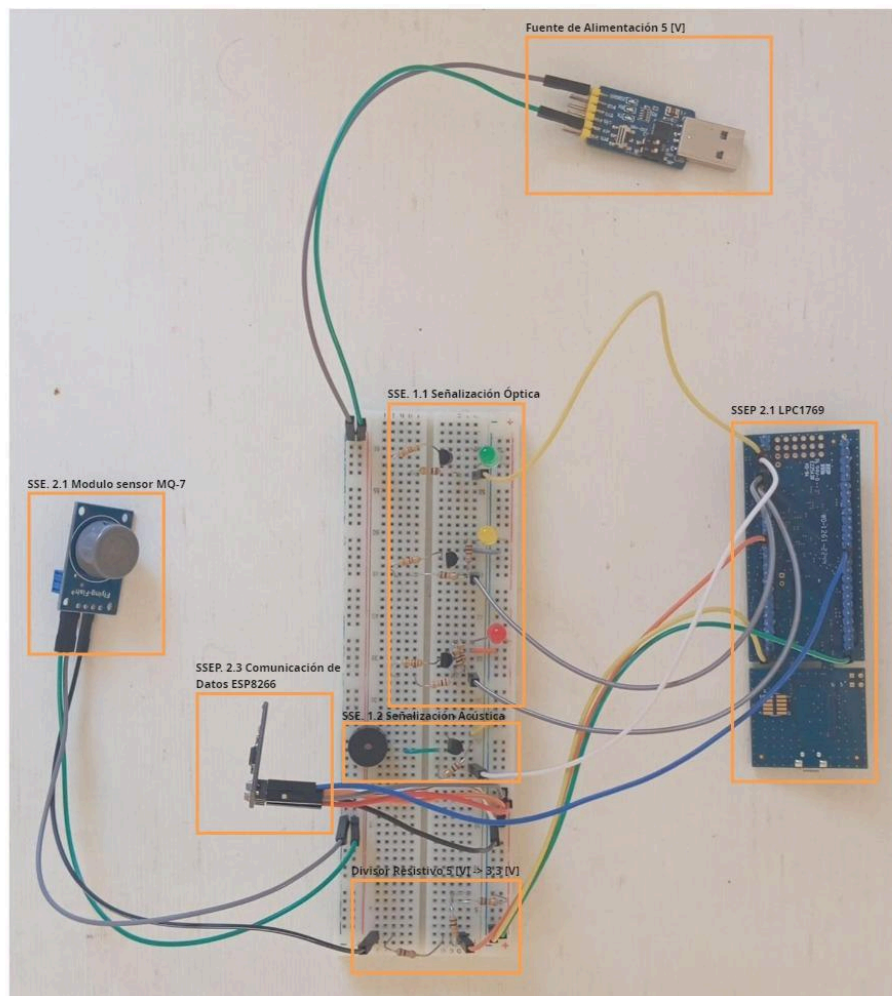
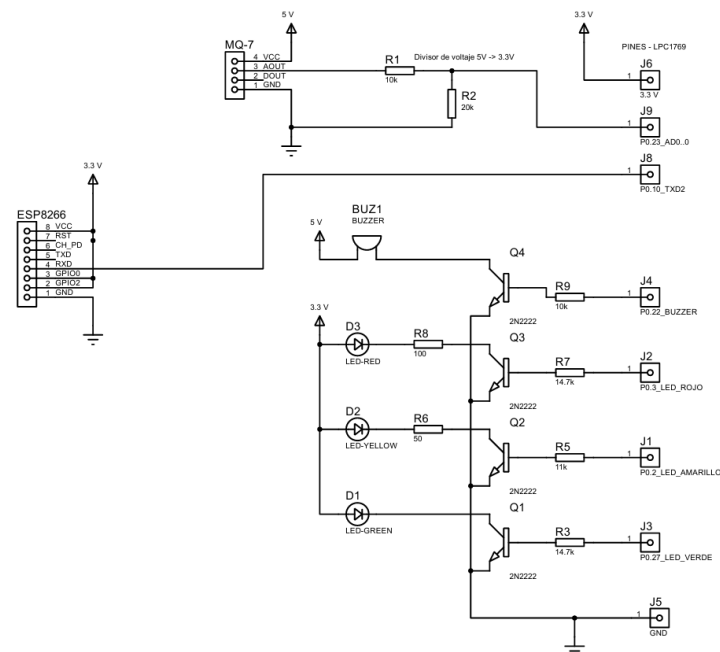
La idea sobre la cual se desarrolló el trabajo, tiene como foco el monitoreo de gas en dispositivos emisores del mismo, de forma tal de poder detectar fugas no deseadas que puedan resultar riesgosas para la salud. Por lo tanto, como primera mejora a realizar, se considera un actuador que pudiese regular el paso de monóxido de carbono del emisor, para que llegado al caso de estar en una situación de alerta, el sistema pueda cerrar el paso de forma autónoma.

7. BIBLIOGRAFÍA

1. HIGIENE Y SEGURIDAD EN EL TRABAJO. Resolución 295/2003. Recuperado de <https://www.argentina.gob.ar/normativa/nacional/resoluci%C3%B3n-295-2003-90396/texto>
2. Centers for Disease Control and Prevention. NIOSH Pocket Guide to Chemical Hazards – Carbon monoxide. Recuperado de <https://www.cdc.gov/niosh/npg/npgd0105.html>
3. How to Use the MQ-7 Carbon Monoxide Sensor. Recuperado de <https://www.teachmemicro.com/use-mq-7-carbon-monoxide-sensor/>
4. LPC1769 USER MANUAL. Recuperado de https://drive.google.com/file/d/1BxeQVesQgAabMhWhEc_bcC6LzO5qRXpJ/view?usp=drive_link

8. ANEXO

8.1. CIRCUITO ESQUEMÁTICO E IMPLEMENTACIÓN DEL SEP



8.2. FIRMWARE DEL SSEP LPC1769

[FIRMWARE LPC1769](#)

8.3. FIRMWARE DEL SSEP ESP8266

[FIRMARE ESP8266](#)

8.4. HOJAS DE DATOS

[MQ-7](#)

[ESP-01](#)

[LPC1769](#)

[BUZZER](#)

[2N2222A](#)