



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

Dispositivo wifi para la monitorización y control del
monóxido de carbono

Trabajo Fin de Grado

Grado en Ingeniería Electrónica Industrial y Automática

AUTOR/A: Arenas Aparisi, Antonio

Tutor/a: Rodríguez Ballester, Francisco

CURSO ACADÉMICO: 2023/2024

Resumen

Respirado en altas concentraciones, el monóxido de carbono es perjudicial para la salud. Puesto que este gas puede llegar a generarse con facilidad en los hogares, es importante controlar sus niveles para proteger a las personas. Por tanto, se va a diseñar un aparato domótico capaz de detectar esta sustancia tóxica y actuar de forma acorde. Este estará construido a partir de un microcontrolador, que gestionará el funcionamiento del resto de componentes, entre ellos los sensores de CO y temperatura. Asimismo, contará con capacidades de comunicación inalámbrica para poder conectarse vía wifi a un servidor abierto en una aplicación propia desde la cual se podrán visualizar las mediciones. Además, se elaborará una maqueta que simulará el funcionamiento de un ventilador que se activa cuando los niveles de monóxido de carbono superan un valor umbral de seguridad para los humanos. Dicho modelo estará compuesto por un motor, cuya velocidad de giro variará según la concentración de gas mediante el uso de un controlador de motores, y por un codificador rotatorio, que se empleará para medirla.

Palabras clave: microcontrolador, sistema embebido, monóxido de carbono, wifi, sensor, extracción de aire.

Abstract

When breathed in high concentrations, carbon monoxide is harmful to health. Since this gas can be easily generated in homes, it is important to control its levels to protect people. Therefore, a home automation device capable of detecting this toxic substance and act accordingly is going to be designed. It will be built from a microcontroller, which will control the operation of the other components, including the CO and temperature sensors. It will also have wireless communication capabilities to be able to connect via wifi to an open server in a particular application from which measurements could be visualized. In addition, a model will be developed to simulate the operation of a fan that is activated when carbon monoxide levels exceed a threshold value for human safety. This model will consist of a motor, whose rotational speed can be varied depending on the gas concentration by using a motor controller, and a rotary encoder, which will be used to measure it.

Keywords: microcontroller, embedded system, carbon monoxide, wifi, sensor, air extraction.

Resumen

Respirat en altes concentracions, el monòxid de carboni és perjudicial per a la salut. Com este gas pot arribar a generar-se amb facilitat en les llars, és important controlar els seus nivells per a protegir les persones. Per tant, es va a dissenyar un aparell domòtic capaç de detectar esta substància tòxica i actuar de manera concorde. Este estarà construït a partir d'un microcontrolador, que gestionarà el funcionament de la resta de components, entre ells els sensors de CO i temperatura. A més, comptarà amb capacitats de comunicació sense cable per a poder connectar-se via wifi a un servidor obert en una aplicació pròpia des de la qual es podran visualitzar els mesuraments. A més, s'elaborarà una maqueta que simularà el funcionament d'un ventilador que s'activa quan els nivells de monòxid de carboni superen un valor límit de seguretat per als humans. Este model estarà compost per un motor, la velocitat de gir del qual es podrà variar segons la concentració de gas mitjançant l'ús d'un controlador de motors, i per un codificador rotatiu, que s'emprarà per a mesurar-la.

Paraules clau: microcontrolador, sistema embegut, monòxid de carboni, wifi, sensor, extracció d'aire.

Índice general

1. MEMORIA	1
1 OBJETO.....	5
2 ANTECEDENTES.....	5
3 ESTUDIO DE NECESIDADES.....	9
4 ESTUDIO DE SOLUCIONES ALTERNATIVAS.....	14
5 DESCRIPCIÓN DE LA SOLUCIÓN ADOPTADA.....	48
6 DESARROLLO DEL PROYECTO. IMPLEMENTACIÓN.....	74
7 PRUEBA Y ANÁLISIS DE RESULTADOS.....	131
8 CONCLUSIONES.....	139
9 REFERENCIAS.....	141
ANEXO I: OBJETIVOS DE DESARROLLO SOSTENIBLE.....	147
2. PLANOS	149
3. PLIEGO DE CONDICIONES	157
1 OBJETO.....	159
2 NORMATIVA DE CARÁCTER GENERAL.....	159
3 MATERIALES.....	159
4 CONDICIONES DE LA EJECUCIÓN.....	162
5 PRUEBAS DE SERVICIO.....	163
6 PRUEBA GENERAL.....	166
4. PRESUPUESTO	169
1 TABLA DE PRECIOS ELEMENTALES.....	171
2 CUADRO DE PRECIOS DESCOMPUESTOS.....	171
3 ESTADO DE MEDICIONES.....	173
4 RESUMEN DEL PRESUPUESTO.....	173

Índice de figuras

Figura 1. Ilustración de una casa automatizada.....	6
Figura 2. Detector de monóxido de carbono a pilas comercial.....	8
Figura 3. Alternativa 1 de monitorización de CO: Sensor de gas MQ-7.....	15
Figura 4. Alternativa 1 de monitorización de CO: Módulo que incluye el sensor MQ-7.....	16
Figura 5. Características de sensibilidad típicas del sensor MQ-7 para distintos gases [7].....	16
Figura 6. Alternativa 2 de monitorización de CO: Sensor de gas MQ-9.....	17
Figura 7. Características de sensibilidad típicas del sensor MQ-9 para distintos gases [8].....	18
Figura 8. Alternativa 3 de monitorización de CO: Sensor de gas TGS5141.....	19
Figura 9. Diferencias entre algunos de los sensores de CO de la compañía Figaro Engineering Inc.....	19
Figura 10. Características de sensibilidad típicas del sensor TGS5141 para distintos gases.....	20
Figura 11. Alternativa 3 de monitorización de CO: Módulo que incluye el sensor TGS5141.....	20
Figura 12. Logo tecnología LoRa.....	21
Figura 13. Logo Bluetooth.....	23
Figura 14. Logo Arduino.....	26
Figura 15. Alternativa 1 de control: Arduino nano.....	27
Figura 16. Distribución de pines del Arduino nano.....	28
Figura 17. Alternativa 2 de control: Arduino Uno R4 WiFi.....	29
Figura 18. Alternativa 3 de control: ESP32-DevKitC-32E.....	31
Figura 19. Logo ST.....	31
Figura 20. STM32 32-bit Arm Cortex-M MCUs.....	33
Figura 21. Comparativa boards STM32.....	35
Figura 22. Alternativa 4 de control: Placa STM32L053R8.....	37
Figura 23. Control todo/nada sin y con histéresis.....	39
Figura 24. Dos señales PWM con ciclos de trabajo distintos.....	40
Figura 25. Ventilador axial Ebm-papst W3G300-UL11-35.....	41
Figura 26. Características técnicas ventilador W3G300-UL11-35.....	42
Figura 27. Motor DC Nidec serie DMN29, 12 V.....	43
Figura 28. Placas de expansión STM32 Nucleo para el control de motores.....	44
Figura 29. Placas de expansión STM32 Nucleo para el control de motores DC con escobillas y sus características.....	45
Figura 30. Alternativa 1 para el control de motores: Placa X-NUCLEO-IHM13A1.....	45
Figura 31. Plano motor DMN29.....	46
Figura 32. Codificadores rotatorios AMT102 y AMT103 de CUI Devices.....	47
Figura 33. Diagrama de flujo funcionamiento del dispositivo (elaboración propia).....	48
Figura 34. Estructura desglosada del proyecto (elaboración propia).....	49
Figura 36. Placa STM32L053R8 junto al módulo de expansión P-NUCLEO-IKA02A1.....	51

Figura 37. Características básicas del sensor de gas TGS5141 [10].....	52
Figura 38. Características eléctricas del sensor analógico de temperatura STLM20 [19]	53
Figura 39. Valores nominales máximos del sensor analógico de temperatura STLM20 [19]	53
Figura 40. Tensión de salida del sensor STLM20 respecto a la temperatura [19]	54
Figura 41. Valores de funcionamiento de la placa STM32L053R8 [20]	54
Figura 42. Funcionalidades de la placa STM32L053R8 dependiendo de su rango de tensión de alimentación [20]	55
Figura 43. Funcionalidades, componentes y periféricos de la placa STM32L053R8 [20].....	56
Figura 44. Esquema funcionamiento comunicación USART/UART (elaboración propia)	56
Figura 45. Logo KEIL (compañía)	58
Figura 46. Chip ESP8266EX	59
Figura 47. Conexiones módulo ESP01s	60
Figura 47. Listado de algunos de los comandos AT básicos para manejar el chip ESP8266 [25]	62
Figura 49. Listado de algunos comandos AT para controlar las funcionalidades wifi del chip ESP8266 [22].....	62
Figura 50. Ejemplo de respuesta al comando AT+CIFSR de un chip ESP266 conectado en modo punto de acceso	65
Figura 51. Logo framework QT	69
Figura 52. Esquema conexiones módulo X-NUCLEO-IHM13A1 [28]	71
Figura 53. Configuración puentes de soldadura P-NUCLEO-IKA02A1 [18]	74
Figura 54. Colocación del jumper JP1 del módulo P-NUCLEO-IKA02A1 para los distintos sensores compatibles [29]	75
Figura 55. Colocación del jumper JP2 del módulo P-NUCLEO-IKA02A1 para ajustar la ganancia [29]	75
Figura 56. Circuito de amplificación y conversión a voltaje de la señal proveniente del sensor TGS5141 (elaboración propia)	76
Figura 57. Coeficientes de compensación del CO con la temperatura del sensor TGS5141 [30]	77
Figura 58. Dependencia de la temperatura en la medida de CO del sensor TGS5141.....	78
Figura 59. Salida del debugger de Keil para una medición de CO en un ambiente sin presencia de este gas.....	80
Figura 60. Consumo de corriente del chip ESP8266EX en sus diferentes modos de operación [31]	80
Figura 61. Características de corriente de la placa STM32L053R8 [20]. Subrayado el valor de salida de corriente de los pines de la placa.	81
Figura 62. Adaptador convertidor DC 9V 1A	82
Figura 63. Módulo de fuente de alimentación de 3,3 V y 5 V conectado a una protoboard	83
Figura 64. Módulo USB-TTL CH340G y cables Dupont	84
Figura 65. Configuración nueva conexión Tera Term	85
Figura 66. Configuración y conexión del puerto serie Tera Term	86
Figura 67. Configuración terminal Tera Term.....	86
Figura 68. Envío y respuesta de los comandos AT y AT+RST al chip ESP8266 vistos desde un terminal abierto en Tera Term	87

Figura 69. Envío y respuesta del comando AT+CWMODE al chip ESP8266 visto desde un terminal abierto en Tera Term	87
Figura 70. Envío y respuesta del comando AT+CWLAP al chip ESP8266 visto desde un terminal abierto en Tera Term.....	87
Figura 71. Ilustración de la comunicación USART/UART entre el microcontrolador STM32L0 y el módulo wifi ESP-01s (elaboración propia).....	88
Figura 72. Tabla de las funciones alternas de los pines GPIO del puerto A del microcontrolador STM32L0 [20].	89
Figura 73. STM32L053R8 Morpho header CN7	89
Figura 74. Rango tensión alimentación STM32L0 [20].....	90
Figura 75. Ilustración conexas del módulo wifi ESP01s, placa STM32L053R8 y módulo de expansión P-NUCLEO-IKA02A1 (elaboración propia)	91
Figura 75. Resumen características motor Nidec DMN29BA [16]	91
Figura 76. Visión general de las partes y componentes de la placa X-Nucleo-IHM13A1	92
Figura 78. Esquema del circuito de la placa X-NUCLEO-IHM13A1 [28]	93
Figura 79. STM32L053R8 características y periféricos. Subrayados los timers de la placa.	93
Figura 80. STM32L053R8 Arduino header CN9	94
Figura 81. Tabla de las funciones alternas de los pines GPIO del puerto B del microcontrolador STM32L0 [20].	95
Figura 82. Comparación timers STM32L053R8 [20]	95
Figura 83. Circuito de control por PWM ventilador W3G300-UL11-35 [16].....	96
Figura 84. Generación de una señal PWM con la salida de un contador	97
Figura 85. Placa X-NUCLEO-IHM13A1 conectada a una STM32 board para el control de un motor DC.....	98
Figura 86. Manual de montaje del encoder AMT103 en un motor [33]	99
Figura 87. Señales de salida del codificador rotatorio [34]	100
Figura 88. Dispositivo montado.....	101
Figura 89. CubeMX selección de la placa.....	102
Figura 90. Implementación tecnología USART en el microcontrolador STM32L0 [20].....	103
Figura 91. CubeMX selección pines UART.....	103
Figura 92. CubeMX parámetros pines UART.....	104
Figura 93. CubeMX Selección pin Canal 1 TIM22.....	104
Figura 94. CubeMX Activación TIM6.....	105
Figura 95. CubeMX selección pines salida digital control del driver.....	105
Figura 96. CubeMX interrupción pin C8.....	106
Figura 97. CubeMX configuración final pines del microcontrolador.....	106
Figura 98. Esquema de la velocidad de giro del motor según la cantidad de CO medido. Elaboración propia. .	115
Figura 99. Diagrama de flujo del funcionamiento de la función proceso_Completo()	117
Figura 100. Diagrama de flujo del funcionamiento de la función main()	120
Figura 101. Diagrama de flujo del funcionamiento de la aplicación diseñada en QT para recibir datos del dispositivo	121

<i>Figura 102. Diagrama de transmisión de datos dentro de las clases utilizadas en la aplicación de visualización</i>	126
<i>Figura 103. Bibliotecas usadas en la aplicación COserver</i>	127
<i>Figura 104. Interfaz gráfica: primera pestaña, Servidor</i>	128
<i>Figura 105. Interfaz gráfica: segunda pestaña, Medidas</i>	129
<i>Figura 106. Interfaz gráfica: ventana gráfico de línea</i>	129
<i>Figura 107. Prueba de funcionamiento: pestaña inicial de la aplicación, sin iniciar la comunicación</i>	131
<i>Figura 108. Prueba de funcionamiento: Terminal abierto en Tera Term y mensajes enviados por el microcontrolador</i>	132
<i>Figura 109. Prueba de funcionamiento: pestaña Servidor. Puerto abierto a la comunicación y conexión del chip wifi al servidor</i>	133
<i>Figura 110. Proceso para probar niveles de CO altos en la placa P-NUCLEO-IKA02A1</i>	134
<i>Figura 111. Prueba de funcionamiento: pestaña de Medidas cuando el monóxido de carbono detectado no supera el valor límite seguro</i>	135
<i>Figura 112. Prueba de funcionamiento: pestaña de Medidas cuando el monóxido de carbono detectado supera el valor límite seguro y se activan la alarma y el motor</i>	135
<i>Figura 113. Prueba de funcionamiento: ventana con un gráfico que representa el valor medido de monóxido de carbono con respecto al tiempo</i>	136
<i>Figura 114. Prueba de funcionamiento: ventana con un gráfico que representa el valor medido de monóxido de carbono con respecto al tiempo (vista ampliada)</i>	137

Índice de tablas

<i>Tabla 1: Efectos sobre la salud de la exposición a distintas concentraciones de monóxido de carbono. Consejería de Salud y Política Social de la Región de Murcia.....</i>	<i>10</i>
<i>Tabla 2: Resumen de las necesidades funcionales del dispositivo.....</i>	<i>12</i>
<i>Tabla 3: Resumen de las características de cada clase de Bluetooth.....</i>	<i>23</i>
<i>Tabla 4: Propiedades y comparación de las tecnologías inalámbricas seleccionadas.....</i>	<i>24</i>
<i>Tabla 5: Tipos de comandos AT del chip ESP8266.....</i>	<i>61</i>
<i>Tabla 6: Resumen comando AT+RST del chip ESP8266.....</i>	<i>62</i>
<i>Tabla 7: Resumen comando AT+CWMODE_CUR del chip ESP8266.....</i>	<i>63</i>
<i>Tabla 8: Resumen comando AT+CWJAP del chip ESP8266.....</i>	<i>64</i>
<i>Tabla 9: Resumen comando AT+CWQAP del chip ESP8266.....</i>	<i>64</i>
<i>Tabla 10: Resumen comando AT+CIFSR del chip ESP8266.....</i>	<i>65</i>
<i>Tabla 11: Resumen comando AT+CIPSTART del chip ESP8266.....</i>	<i>65</i>
<i>Tabla 12: Resumen comando AT+CIPMODE del chip ESP8266.....</i>	<i>66</i>
<i>Tabla 13: Resumen comando AT+CIPSEND del chip ESP8266.....</i>	<i>66</i>
<i>Tabla 14. Listado de bases de datos compatibles con QT y el nombre del driver para manejarlas.....</i>	<i>70</i>
<i>Tabla 15: Resumen conexión entre la board y el módulo wifi.....</i>	<i>90</i>
<i>Tabla 16: Resumen de la asignación de los pines GPIO de la board STML053R8.....</i>	<i>100</i>
<i>Tabla 17. Definición etiquetas Figura 105. Dispositivo montado.....</i>	<i>101</i>
<i>Tabla 18. Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).</i>	<i>147</i>

Siglas:

CO	Monóxido de carbono
IoT	<i>Internet of Things</i>
UART	<i>Universal Asynchronous Receiver / Transmitter</i>
ppm	Partes por millón
CPU	Central Processing Unit
SPI	<i>Serial Peripheral Interface</i>
SDIO	<i>Secure Digital Input Output</i>
I ² C	<i>Inter-Integrated Circuit</i>
RAM	<i>Random Access Memory</i>
ROM	<i>Read Only Memory</i>
PWM	<i>Pulse Width Modulation</i>
UNE	Una Norma Española
RPM	Revoluciones por minuto
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>

1. MEMORIA

Índice Memoria

1 OBJETO	5
2 ANTECEDENTES	5
3 ESTUDIO DE NECESIDADES	9
3.1 MEDICIÓN DE GAS MONÓXIDO DE CARBONO	9
3.2 CAPACIDAD DE CONTROL.....	11
3.3 TRANSMISIÓN Y VISUALIZACIÓN DE DATOS.	11
3.4 MEDIDAS DE SEGURIDAD ANTE LA PRESENCIA DE CO	11
3.5 RESUMEN DE LAS NECESIDADES FUNCIONALES	12
4 ESTUDIO DE SOLUCIONES ALTERNATIVAS	14
4.1 MEDICIÓN DEL MONÓXIDO DE CARBONO.	14
4.1.1 Sensor de gas MQ-7	14
4.1.2 Sensor de gas MQ-9	17
4.1.3 Sensor de CO TGS5141	18
4.2 TRANSMISIÓN Y VISUALIZACIÓN DE DATOS.	21
4.2.1 LoRa.....	21
4.2.2 Wifi.....	22
4.2.3 Bluetooth.....	22
4.2.4 Comparación y selección de tecnologías de transmisión datos.	23
4.3 CONTROL	25
4.3.1 Arduino.....	26
4.3.2 ESP32.....	29
4.3.3 STM32	31
4.4 MEDIDAS DE SEGURIDAD ANTE LA PRESENCIA DE CO.	37
4.4.1 Extracción de aire.....	38
5 DESCRIPCIÓN DE LA SOLUCIÓN ADOPTADA	48
5.1 SISTEMA DE MEDICIÓN DEL MONÓXIDO DE CARBONO.	50
5.2 SISTEMA ELECTRÓNICO DE CONTROL Y TRANSMISIÓN DE DATOS.	54
5.2.1 Microcontrolador	54
5.2.2 Módulo wifi	59
5.3 VISUALIZACIÓN DE LA INFORMACIÓN E INTERFAZ GRÁFICA	67
5.3.1 TCP/IP.....	67
5.3.2 QT Creator.....	69
5.3.3 SQLite	70

5.4 EXTRACCIÓN DE AIRE	71
5.4.1 Controlador y motor	71
5.4.2 Codificador rotatorio	72
6 DESARROLLO DEL PROYECTO. IMPLEMENTACIÓN	74
6.1 CÁLCULOS Y MONTAJE DEL HARDWARE	74
6.1.1 Conexión y configuración del sensor de CO	74
6.1.2 Integración de la conectividad wifi	80
6.1.3 Elaboración maqueta ventilador	91
6.1.4 Resumen montaje y conexiones	100
6.2 DESARROLLO DEL SOFTWARE	101
6.2.1 Programación MCU	102
6.2.2 Visualización de la información. Servidor e interfaz gráfica	121
7 PRUEBA Y ANÁLISIS DE RESULTADOS	131
8 CONCLUSIONES	139
9 REFERENCIAS	141
ANEXO I: OBJETIVOS DE DESARROLLO SOSTENIBLE	147

1 OBJETO

El presente proyecto consiste en el diseño de un dispositivo electrónico basado en un microcontrolador para la monitorización y control en una estancia de la concentración de monóxido de carbono, un gas perjudicial para la salud humana, así como la activación de una respuesta en caso de que se sobrepase un nivel seguro.

Se ha planteado una maqueta que simula el funcionamiento mediante sensores de CO y temperatura, así como un motor cuyo control está basado en el de un ventilador industrial real. Además, el sistema contará con conexión wifi de manera que se pueda conectar a un PC y se puedan leer las mediciones del sensor desde un servidor abierto desde una aplicación.

La parte de diseño correspondiente a este trabajo es la selección de la electrónica a utilizar, la proposición de montaje de la maqueta y la programación necesaria del microcontrolador para el funcionamiento.

2 ANTECEDENTES

Durante las últimas décadas, la electrónica de consumo se ha ido introduciendo en nuestros hogares de maneras muy diversas con el objetivo de facilitar tareas y obligaciones diarias. La rápida expansión del uso de dispositivos electrónicos en el domicilio, particularmente debido a su reducción en tamaño y precio, así como aumento de funcionalidades, ha revolucionado la manera en que vivimos. Este es el fin en el que se centra el campo la domótica.

La domótica es el conjunto de sistemas utilizados para automatizar una vivienda, normalmente centrados en la gestión energética, aumento del bienestar, la seguridad, la accesibilidad y las comunicaciones [1]. Se puede definir como el uso de la tecnología para controlar y gestionar una residencia de forma automática e inteligente.

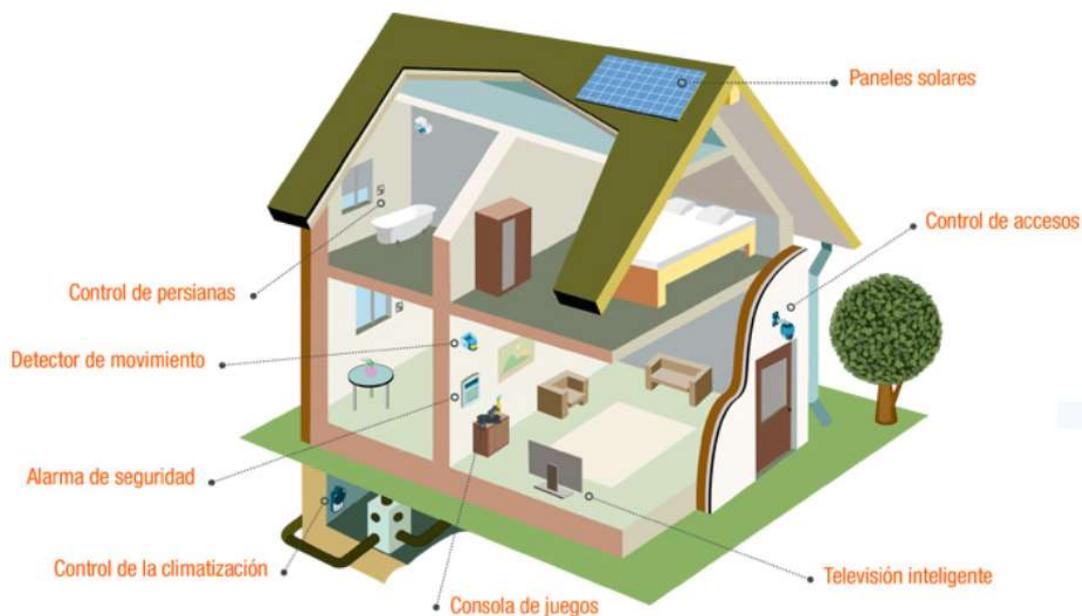


Figura 1. Ilustración de una casa automatizada.¹

En definitiva, el objetivo fundamental de la domótica es el de mejorar el día a día de las personas, facilitando labores diarias en el hogar y aumentando la calidad de vida. Para ello, podemos encontrar diferentes herramientas, como puertas automáticas, ventilación y climatización controlada por dispositivos móviles, alarmas y cámaras de seguridad con conexión wifi o persianas controladas inalámbricamente. Pero no todo es comodidad, otra de sus principales características es su capacidad de apoyar en la seguridad de los habitantes de una vivienda. Desde hace años, es común encontrar en hogares algún dispositivo para cuidar de la salud de las personas que ahí viven, como los detectores de humo que lanzan avisos en caso de incendio o sensores que comunican a una aplicación si se detectan gases peligrosos, como por ejemplo el monóxido de carbono.

Y es que, aunque posiblemente menos discutido y comentado en la actualidad que el dióxido de carbono, el monóxido de carbono es realmente peligroso para los humanos. Según la SEPAR, Sociedad Española de Neumología y Cirugía Torácica, entre los contaminantes más peligrosos para la salud se encuentran el nitrógeno, el ozono, las partículas en suspensión, el dióxido de azufre, el monóxido de carbono y el plomo [2].

El problema con este gas es que, cuando se inhala, sustituye al oxígeno en la hemoglobina de la sangre y esta mezcla da como resultado una proteína conocida como carboxihemoglobina (COHb). Esto provoca que haya menos hemoglobina disponible para

¹ Extraída de: <https://www.iberdrola.com/innovacion/domotica>

transportar oxígeno desde los pulmones a los tejidos del cuerpo, provocando un estado de hipoxia.

Al no tener olor ni color, puede afectar a la persona que lo esté respirando sin que esta lo perciba. Por ello, es importante conocer los síntomas de la hipoxia y del envenenamiento por este gas; la exposición a concentraciones bajas suele causar dolores de cabeza y sensación de ahogo al hacer ejercicio, mientras que una exposición más continuada puede provocar síntomas similares a los de la gripe, dolores de cabeza, mareos, náuseas, cansancio y confusión [3]. En casos muy graves puede causar la pérdida de consciencia, daño cerebral irrecuperable, problemas de corazón e incluso la muerte. Además, después de la recuperación de la intoxicación pueden aparecer otros síntomas, incluso cuando ya no hay presencia del gas, como la pérdida de memoria, cambios de personalidad o problemas de movimiento.

Según varios artículos de prensa que recogen un estudio realizado por la SEPAR, en España la media anual es de entre 5.000 y 10.000 intoxicaciones por este gas, con unos 125 casos que acaban en muerte [4] [5] [6].

A pesar de sus peligros, el CO es relativamente común hoy en día, ya que se produce en las malas combustiones del gas natural, propano, petróleo, queroseno, madera y carbón (por lo general, aquellos combustibles que contengan átomos de carbono y que no dispongan de suficiente oxígeno en la combustión como para formar CO₂ pueden ser una fuente de CO). La mayor parte de los casos de envenenamiento por monóxido de carbono se da en los hogares, sobre todo en actividades diarias como cocinar y usar la calefacción, sobre todo si se utilizan estufas, chimeneas o calderas de calefacción y agua antiguas o en mal estado [3]. Para evitar desgracias, por tanto, es vital monitorizar este gas en espacios cerrados.

Los detectores domésticos de monóxido de carbono son una realidad desde hace años por ese motivo. Se trata de dispositivos que miden continuamente los niveles de este gas en el aire y generan algún tipo de alerta si se alcanzan niveles nocivos: desde simples alarmas sonoras en los más básicos hasta notificaciones en teléfonos móviles en aquellos más modernos. En muchos casos, además, estos se pueden encontrar junto a detectores de humo o instalados en casas con diversos sistemas domóticos. Son especialmente útiles en hogares que cuentan con chimeneas, calefactores o aparatos que queman combustible, como estufas o cocinas de gas, pues son aquellos que pueden llegar a generar CO.



Figura 2. Detector de monóxido de carbono a pilas comercial.²

Por ello, este proyecto consistirá en la realización de un dispositivo de prestaciones similares a aquellos detectores de monóxido de carbono que se pueden encontrar en el mercado, diseñando la electrónica, programando su funcionamiento y dotándolo de conexión wifi, para que, finalmente, pudiera llegar a considerarse un producto domótico de consumo. Además, sumado a eso, se planteará una maqueta que simulará el giro de un ventilador extractor que se activa para renovar el aire de una estancia en la que los niveles de CO son nocivos para el ser humano.

² Extraída de: <https://www.cdc.gov/spanish/nceh/especiales/monoxidodecarbono/index.html>

3 ESTUDIO DE NECESIDADES

El primer paso en el desarrollo del proyecto consiste en plantearse cuáles son sus necesidades funcionales, es decir, determinar qué se quiere que haga y qué requiere para hacerlo.

Para ello, antes de establecerlas, es buena idea entender cómo funcionan los detectores de monóxido de carbono que se pueden encontrar en el mercado y qué hacer para diseñar uno propio. Una búsqueda rápida en diversas páginas de venta de productos por internet muestra que la gran mayoría de ellos tienen características similares, entre ellas, la capacidad de detección de dicho gas, control por microprocesador, algún tipo de alarma sonora, bajo precio y tamaño reducido.

Una vez visto lo que ofrecen otros aparatos con la misma finalidad, se determina que el dispositivo debe de ser capaz de tomar una medida de la cantidad de monóxido de carbono dentro de una estancia con precisión y de transmitir estos datos para actuar de forma acorde.

Las necesidades funcionales, por tanto, son las siguientes:

- Medición de gas monóxido de carbono.
- Capacidad de control.
- Transmisión y visualización de datos.
- Posibilidad de extracción de aire.

3.1 Medición de gas monóxido de carbono

Como se ha comentado anteriormente, el monóxido de carbono es un gas perjudicial para la salud humana ya que es capaz introducirse en la sangre cuando es inhalado, impidiendo que el oxígeno necesario para la vida entre al organismo y pudiendo llegar a causar daños en los tejidos e incluso la muerte. Por eso mismo, es especialmente importante contar con un sensor preciso y fiable, que sea capaz de detectar las proporciones nocivas de este gas en el aire.

Una opción para considerar los valores límites peligrosos es tomar en consideración las regulaciones y normativas vigentes. En España, de acuerdo con el Real Decreto 102/2011, de 28 de enero, relativo a la mejora de la calidad del aire, el valor límite de la media de ocho horas máxima en un día para la protección de la salud humana es de 10 mg/m³ de CO [7].

De igual manera, la EPA, Agencia de Protección Ambiental de Estados Unidos, establece el límite de monóxido de carbono en el aire también en 10 mg/m³ para una media de ocho

horas, mientras que los máximos establecidos por la Administración de Seguridad y Salud Ocupacional de dicho país (OSHA) son de 55 mg/m³ durante una jornada de trabajo [8].

También es vital considerar los efectos en la salud humana a la hora de ajustar los parámetros, incluso aquellos que pueden ocurrir para concentraciones de CO ambiental menores a las reguladas. Para ello, en la tabla siguiente, tomada de la página web de la Consejería de Salud y Política Social de la Región de Murcia [9] se pueden ver los efectos del monóxido de carbono en diferentes concentraciones:

Concentración de CO	Síntoma
0-229 mg/m ³ (0-200 ppm)	Ligero dolor de cabeza en algunos casos.
10 mg/m ³ (8,7 ppm)	No se excede el nivel carboxihemoglobina del 2,5 %, aun cuando un sujeto normal realiza ejercicio moderado durante 8 horas (siendo el valor estándar de un adulto sano del 1%).
30 mg/m ³ (26 ppm)	No se excede el nivel carboxihemoglobina del 2,5 %, aun cuando un sujeto normal realice ejercicio ligero o moderado durante una hora.
34,4 mg/m ³ (30 ppm)	La exposición diaria a esta concentración es equivalente a fumar 20 cigarrillos al día.
40,1 mg/m ³ (35 ppm)	Límite para personas con enfermedades cardíacas.
60 mg/m ³ (52 ppm)	No se excede el nivel carboxihemoglobina del 2,5 %, aun cuando un sujeto normal realice ejercicio ligero o moderado durante 30 minutos.
100 mg/m ³ (87 ppm)	No se excede el nivel carboxihemoglobina del 2,5 %, aun cuando un sujeto normal realice ejercicio ligero o moderado durante 15 minutos.
115 mg/m ³ (100 ppm)	Primer indicio de angina en sujetos que hacían ejercicio con cardiopatía coronaria expuestos a esta concentración.
229-458 mg/m ³ (200-400 ppm)	De 5 a 6 horas se puede observar un leve dolor de cabeza, náuseas y vértigo.
458-802 mg/m ³ (400-700 ppm)	De 4 a 5 horas se puede observar un fuerte dolor de cabeza, incoordinación muscular, debilidad, vómitos y colapso.
802-1260 mg/m ³ (700-1100 ppm)	De 3 a 5 horas se puede observar un fuerte dolor de cabeza, debilidad, vómitos y colapso.
1260-1832 mg/m ³ (1100-1600 ppm)	De 1,5 a 3 horas hay posibilidad de caer en coma.
1832-2290 mg/m ³ (1600-2000 ppm)	De 1 a 1,5 horas hay posibilidad de muerte.
5726-11452 mg/m ³ (5000-10000 ppm)	De 2 a 15 minutos se puede producir la muerte.

Tabla 1: Efectos sobre la salud de la exposición a distintas concentraciones de monóxido de carbono. Consejería de Salud y Política Social de la Región de Murcia

Por tanto, teniendo en cuenta el valor legislado, así como los niveles críticos para la salud, el dispositivo debe contar con un sensor de monóxido de carbono capaz de medir la concentración de este gas en el ambiente de forma precisa para concentraciones de 10 mg/m³ (8,7 ppm).

3.2 Capacidad de control

Para llevar a cabo la monitorización del CO se necesita recoger las medidas tomadas por los sensores y transformarlas a un valor digital. Por tanto, es preciso contar con un sistema que controle el funcionamiento completo dispositivo, tomando las mediciones y almacenándolas.

En general, para el control del dispositivo se requiere una unidad central de procesamiento, capacidad de memoria y periféricos de entrada y salida. Es decir, un microcontrolador.

3.3 Transmisión y visualización de datos.

Una vez tomadas las medidas se requiere que lleguen de algún modo al usuario. Para ello, este aparato debe contar con la capacidad de enviar información a otro, pues él mismo no dispone de pantalla o algún otro componente de visualización.

Por diseño, el dispositivo debería ser capaz de ser utilizado en diferentes estancias sin limitaciones de dónde colocarlo, por lo que el uso de cables ethernet, por ejemplo, queda excluido. Se determina, por tanto, que la transmisión de datos se debe realizar de forma inalámbrica.

La comunicación ha de ser fiable y funcionar ininterrumpidamente, de modo que se estén tomando y enviando medidas a todas horas. Para mostrar información al usuario, sería interesante hacerlo de modo que se puedan leer las mediciones tomadas desde un lugar diferente al que se encuentra el sensor. Para ello, el proyecto deberá tener la capacidad de conectarse con un PC.

3.4 Medidas de seguridad ante la presencia de CO

De acuerdo con diversos estudios, son comunes los espacios en los que los niveles de CO medidos a lo largo de un día superan los valores límites saludables. En túneles de carretera y aparcamientos subterráneos, donde no hay mucha ventilación en un espacio donde se acumulan muchos motores de combustión, los niveles medios del monóxido de

carbono pueden superar los 115 mg/m³ (100 ppm), con valores pico incluso más altos. Asimismo, en hogares donde se utilizan aparatos de gas, especialmente si tienen algún tipo de avería, pueden darse valores máximos de hasta 60-115 mg/m³ (52-100 ppm), y, en espacios cerrados, el humo del tabaco puede hacer que la concentración media de 8 horas de este gas llegue a alcanzar los 23-46 mg/m³ (20-40 ppm) [9].

Puesto que el monóxido de carbono se ha demostrado perjudicial para la salud humana, es necesario contar con un método para la extracción del aire contaminado de cierta estancia y para el aviso por medios físicos del riesgo al usuario. Dicho sistema será ajeno al aparato, pero ha de ser controlado por señales procedentes del microcontrolador.

3.5 Resumen de las necesidades funcionales

En esta sección se recogen las necesidades que es preciso que el dispositivo sea capaz de cumplir.

Resumen de las necesidades funcionales del dispositivo
<ul style="list-style-type: none"> • Medir la proporción de monóxido de carbono en el aire. Detectar concentraciones mínimas de 8,7 ppm.
<ul style="list-style-type: none"> • Contar con un microcontrolador para la gestión del funcionamiento completo.
<ul style="list-style-type: none"> • Posibilidad de transmisión inalámbrica de información.
<ul style="list-style-type: none"> • Mostrar información al usuario. Capacidad de conexión con un PC externo.
<ul style="list-style-type: none"> • Controlar un sistema externo para la extracción de aire contaminado. Maqueta que simule el funcionamiento real.

Tabla 2: Resumen de las necesidades funcionales del dispositivo

Por otro lado, es necesario conocer la normativa vigente que regula el diseño de este tipo de dispositivos. Por esto, se deben tomar en cuenta las normas redactadas en el Boletín Oficial del Estado (BOE), así como por autoridades competentes tales como la Asociación Española de Normalización (AENOR) o la Comunidad Europea:

- Real Decreto 102/2011, de 28 de enero, relativo a la mejora de la calidad del aire.
- UNE-EN 54-31:2014+A1:2019. Sistemas de detección y alarma de incendios. Parte 31: Detectores de incendios multisensoriales. Detectores puntuales que utilizan una combinación de humo, monóxido de carbono y opcionalmente sensores de calor.
- UNE-EN 50291-1:2018 (Ratificada). Detectores de gas. Aparatos eléctricos para la detección de monóxido de carbono en los locales de uso doméstico. Parte 1: Métodos de ensayo y requisitos de funcionamiento. (Ratificada por la Asociación Española de Normalización en junio de 2018.)
- UNE-EN 50291-2:2020. Aparatos eléctricos para la detección de monóxido de carbono en los locales de uso doméstico. Parte 2: Aparatos eléctricos de

funcionamiento continuo en instalaciones fijas en vehículos de recreo y locales análogos, incluyendo las embarcaciones de recreo. Métodos de ensayo adicionales y requisitos de funcionamiento.

- Real Decreto 110/2015, de 20 de febrero, sobre residuos de aparatos eléctricos y electrónicos.
- UNE-EN 60335-1:2012/A15:2022, Aparatos electrodomésticos y análogos. Seguridad. Parte 1: Requisitos generales.
- Directiva 2003/108/CE del Parlamento Europeo y del Consejo, de 8 de diciembre de 2003, por la que se modifica la Directiva 2002/96/CE sobre residuos de aparatos eléctricos y electrónicos (RAEE).
- Real Decreto 1801/2003, de 26 de diciembre, sobre seguridad general de los productos. BOE-A-2004-511.
- Directiva 2014/35/UE del Parlamento Europeo y del Consejo, de 26 de febrero de 2014, sobre la armonización de las legislaciones de los Estados miembros en materia de comercialización de material eléctrico destinado a utilizarse con determinados límites de tensión.
- Directiva 2014/53/UE del Parlamento Europeo y del Consejo, de 16 de abril de 2014, relativa a la armonización de las legislaciones de los Estados miembros sobre la comercialización de equipos radioeléctricos, y por la que se deroga la Directiva 1999/5/CE.

4 ESTUDIO DE SOLUCIONES ALTERNATIVAS

En esta sección se establecen las diferentes soluciones posibles para cumplir con las necesidades planteadas en el apartado anterior, mostrando las diferentes alternativas de componentes a utilizar para que el proyecto funcione de la forma en que se ha planteado.

En cuanto al criterio de selección empleado, se han tenido en cuenta cinco aspectos relevantes: las características técnicas de cada componente, su tamaño, su coste económico, sus prestaciones y su sencillez en el diseño y montaje. Al seleccionar los distintos elementos del dispositivo el objetivo es encontrar el balance entre cumplir con las especificaciones técnicas mínimas del proyecto, conseguir componentes de mayor fiabilidad y características, evitar un precio final excesivo que desplazara al producto de un hipotético mercado, lograr que el montaje sea lo más simple posible para que no haya errores y conseguir que el aparato final sea reducido y compacto.

Además de estos requerimientos, para cada componente específico se tendrán en cuenta otras virtudes y ventajas técnicas que pueda poseer y que puedan añadir valor al producto final respecto a sus alternativas.

4.1 Medición del monóxido de carbono.

La base del proyecto es la medición de la concentración de monóxido de carbono en el aire, por lo que es vital encontrar un sensor que cumpla esta función correctamente, tenga una sensibilidad adecuada a las necesidades, esté disponible en el mercado con un precio razonable y tenga la posibilidad de ser conectado a la entrada analógica de un controlador con facilidad.

Con estas condiciones se plantearon varias alternativas, entre ellas:

4.1.1 Sensor de gas MQ-7

Este sensor es parte de la familia de sensores analógicos de gases MQ, pensados para medir diversos componentes químicos que se encuentran en el aire.



Figura 3. Alternativa 1 de monitorización de CO: Sensor de gas MQ-7.³

Dentro de esta familia de sensores hay una gran variedad de modelos, diseñados para detectar sustancias como el metano, butano, alcohol, gas natural, hidrógeno, ácido sulfhídrico o el monóxido de carbono, como es el caso del MQ-7.

Su uso está muy extendido, sobre todo en aplicaciones de domótica para la sensorización de gases perjudiciales, proyectos de medición de la calidad del aire y la detección de gases inflamables. Son especialmente comunes debido a su fácil conexión con placas Arduino y su precio reducido. Además, suelen venir en módulos, lo que facilita su uso, puesto que los circuitos ya están hechos y basta con alimentarlos y conectarlos a una placa.

Los sensores de la familia MQ son electroquímicos y funcionan variando su resistencia cuando se encuentran frente al gas para el que están diseñados. Al detectar el gas, la conductividad cambia, variando de este modo el voltaje en el sensor. Con un convertidor analógico-digital se puede transformar esta variación de tensión en una señal digital.

El MQ-7, en particular, es muy sensible al monóxido de carbono y puede detectarlo en el aire a su alrededor en concentraciones de entre las **20 ppm** y **2000 ppm**.

³ Extraída de: <https://store.prometec.net/producto/mq-7-sensor-monoxido-c/>

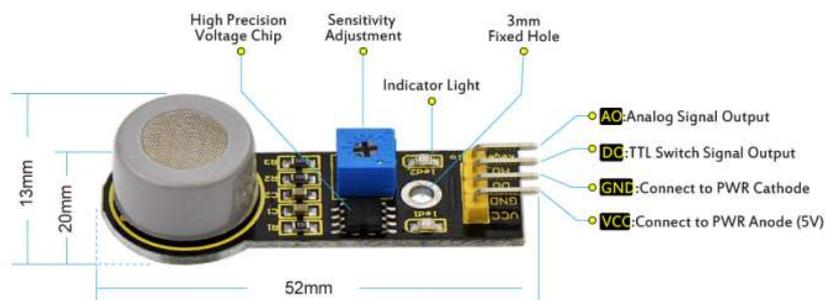


Figura 4. Alternativa 1 de monitorización de CO: Módulo que incluye el sensor MQ-7.⁴

Tiene en su interior un pequeño calentador en forma de resistencia que aumenta la temperatura del sensor electroquímico y permite que los materiales que interaccionan con el gas (SnO_2) obtengan la sensibilidad requerida. Es importante que este calentador alcance la temperatura de funcionamiento para que la lectura sea fiable. Según su *datasheet* [10], este tiempo de precalentado es de mínimo 48 h. Además, necesita una tensión de 1,5 V para alimentar el calentador, a diferencia de otros modelos de la misma familia que precisan de 5 V.

Para obtener los valores de CO en partes por millón, se puede aplicar la siguiente gráfica, que muestra la concentración del gas en ppm a partir de la relación entre la resistencia del sensor R_0 y la resistencia medida R_s (a 20 °C, humedad del 65%, concentración de O_2 del 21 % y R_L de 10 k Ω).

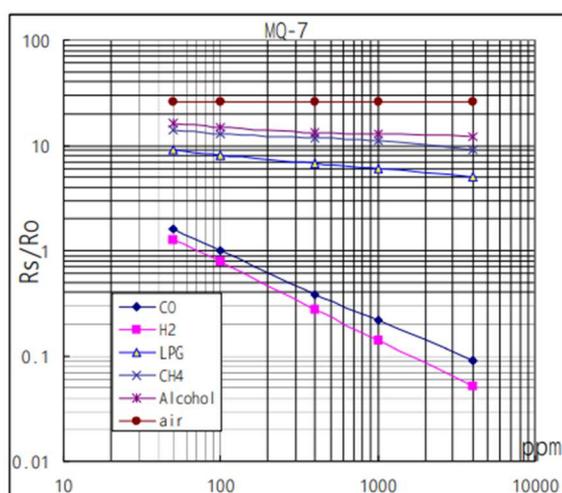


Figura 5. Características de sensibilidad típicas del sensor MQ-7 para distintos gases [10].

⁴ Extraída de: https://wiki.keyestudio.com/Ks0045_keyestudio_MQ-7_Carbon_Monoxide_Gas_Sensor

Entre las ventajas que aportaría este sensor al proyecto están su simplicidad de conexión, su reducido tamaño y precio, respuesta rápida y buena sensibilidad. Sin embargo, en el estudio de necesidades se menciona que el dispositivo debe de ser capaz de medir concentraciones de mínimo 10 ppm, pues tales niveles de CO ya pueden ser perjudiciales en cierta medida para la salud y el bienestar humano. Puesto que el mínimo que es capaz de detectar el sensor MQ-7 es de 20 ppm, no se recomendaría su uso.

4.1.2 Sensor de gas MQ-9

Al igual que el MQ-7, también parte de la familia de sensores de gas MQ, pero con algunas diferencias, principalmente la capacidad de medir otros gases además del monóxido de carbono, como el metano (CH₄) y gases licuados del petróleo (GLP) [11].

Para los diferentes gases utiliza un método de detección cíclica a alta y baja temperatura: se alimenta el calentador a 1,5 V para detectar el monóxido de carbono (baja temperatura) y a 5 V para detectar metano, propano y otros gases licuados.

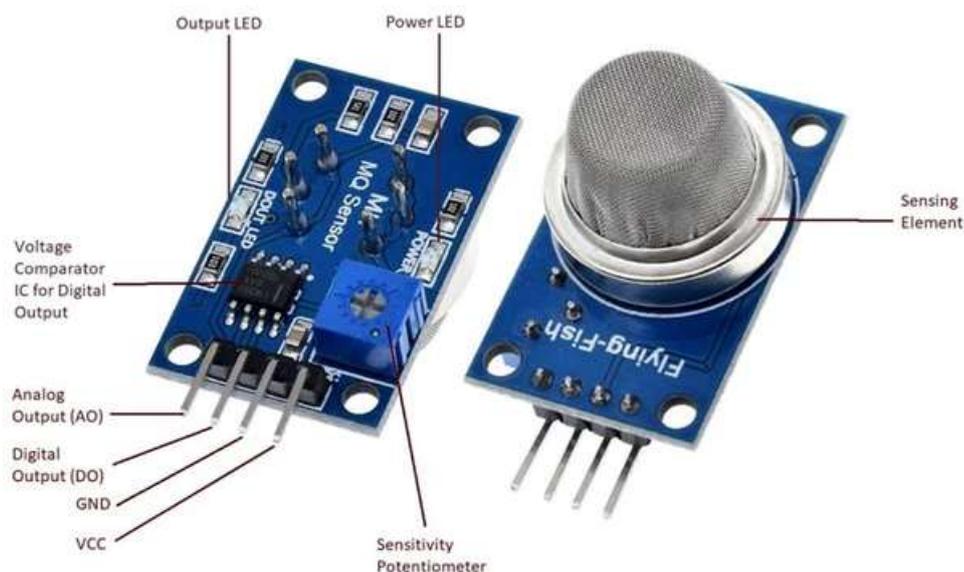


Figura 6. Alternativa 2 de monitorización de CO: Sensor de gas MQ-9⁵

El sensor MQ-9 es válido para varias aplicaciones ya que puede detectar gases que contienen monóxido de carbono y gases combustibles. Por ello, puede ser usado en equipos de detección de fugas de gases inflamables en hogares, automóviles o fábricas.

Funciona de manera similar al sensor MQ-7 anteriormente mencionado: a medida que aumenta la concentración de gas, aumenta la conductividad del sensor electroquímico (y de

⁵ Extraída de: <https://www.makestore.com.au/product/elec-sensor-co-mq-9/>

esta manera la tensión eléctrica). Del mismo modo, también suele encontrarse en módulos de fácil configuración y conexión, por lo que su uso está muy extendido.

Puede detectar el monóxido de carbono en el rango de **10 ppm a 1000 ppm**, con una sensibilidad lineal, como se muestra en la Figura 7, y también el gas licuado del petróleo (GLP) y metano en el de 100 ppm a 10000 ppm.

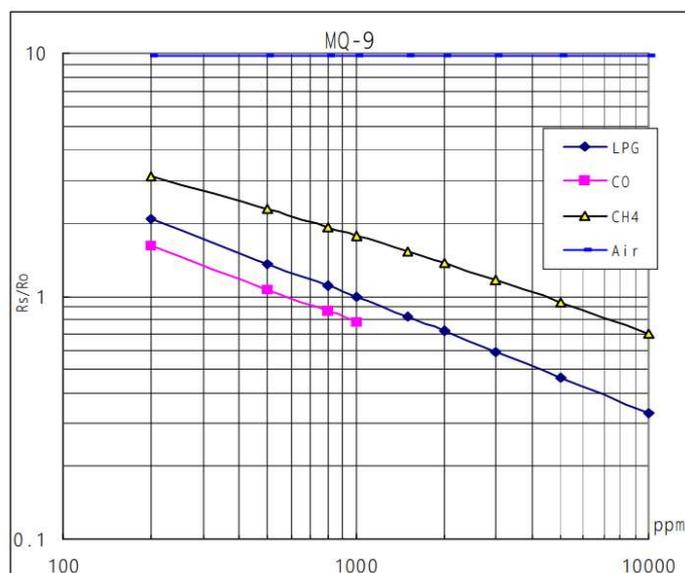


Figura 7. Características de sensibilidad típicas del sensor MQ-9 para distintos gases [8].

Teniendo en cuenta que este sensor comparte las ventajas del MQ-7, pero a diferencia de este sí tiene la posibilidad de detectar concentraciones de CO de 10 ppm, es una mejor opción para este proyecto.

4.1.3 Sensor de CO TGS5141

El TGS5141 es un sensor electroquímico desarrollado por la empresa japonesa *Figaro Engineering INC*, especializada en el desarrollo de sensores de gas semiconductores y electroquímicos, entre otros.



Figura 8. Alternativa 3 de monitorización de CO: Sensor de gas TGS5141⁶

Está diseñado para la medición de monóxido de carbono y para ser acoplado, sobre todo, en alarmas y detectores de este gas, tanto portables como para hogares, así como combinándolo con otros sensores para fabricar detectores de incendios. Esto se debe, principalmente, a su tamaño compacto (50 mm de diámetro y 7 mm de altura), un 10 % del tamaño de su predecesor, el TGS5042. La razón de sus dimensiones es la sustitución de una reserva de agua en el sensor, como la que usa el TGS5042, por un electrolito [12].

	TGS5042	TGS5342	TGS5141
Size	Large	Small	Compact
Expected sensor life	10 yrs+	10 yrs+	10 yrs+
Humidity dependency	no	no	yes (within±5%)
Water reservoir	yes	yes	no

Figura 9. Diferencias entre algunos de los sensores de CO de la compañía Figaro Engineering Inc.

Según su *datasheet* [13], el TGS5141 tiene la capacidad de detectar concentraciones de CO en el aire de entre las **0 ppm y 5000 ppm** aproximadamente, siendo, por tanto, el sensor visto hasta ahora con mejor rango de detección, incluso midiendo valores menores a la decena. Tiene, además, un tiempo de respuesta corto, menor a los 60 segundos.

Por otro lado, sus características cumplen con varias normas establecidas en el estudio de necesidades del proyecto, como la UNE-EN 54-31:2014+A1:2019 de Sistemas de detección y alarma de incendios y la UNE-EN 50291-1:2018 de Aparatos eléctricos para la detección de monóxido de carbono en los locales de uso doméstico.

A continuación, en una gráfica obtenida de la hoja de datos de este sensor [13], se muestra la relación entre su corriente de salida y la concentración de diferentes gases (sensibilidad). Se puede observar que hay una relación lineal con el monóxido de carbono, por lo que este sensor es muy adecuado para medir este gas.

⁶ Extraída de: <https://www.figarosensor.com/product/entry/tgs5141-p00.html>

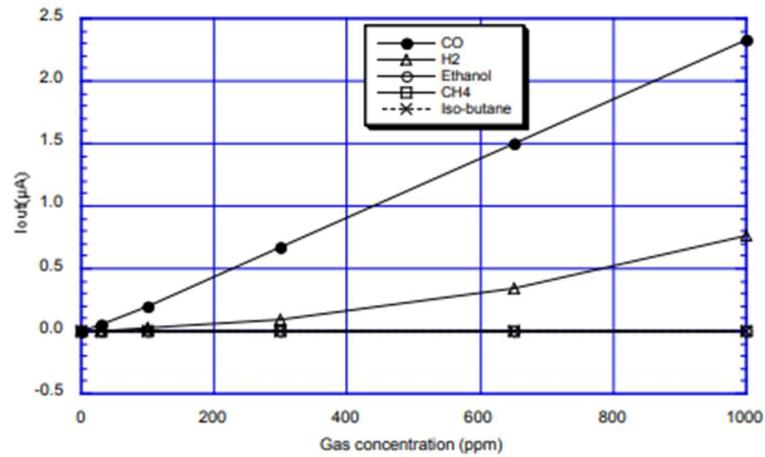


Figura 10. Características de sensibilidad típicas del sensor TGS5141 para distintos gases

Una de sus ventajas es que puede venir incluido en módulos que convierten la salida de corriente del sensor en una señal de tensión y facilitan enormemente la conexión a un microcontrolador.

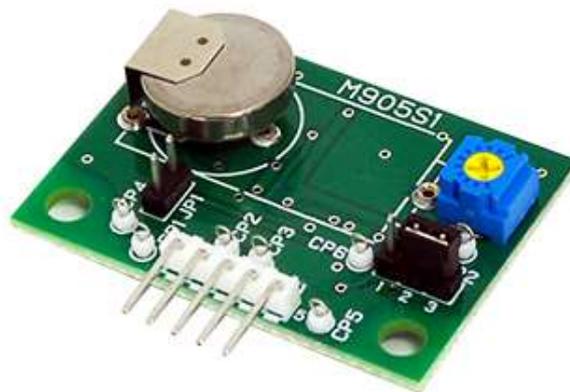


Figura 11. Alternativa 3 de monitorización de CO: Módulo que incluye el sensor TGS5141⁷

Por tanto, al ser este el sensor que mejor rango de detección del CO tiene de entre los comparados (entre las 0 ppm y 5000 ppm), además de un buen tiempo de respuesta y la posibilidad de encontrarlo en varios módulos fáciles de conectar en el mercado, para el proyecto se ha seleccionado finalmente el TGS5141.

⁷ Extraída de: https://www.figarosensor.com/product/entry/em5141a_evaluation_module_for_tgs5141.html

4.2 Transmisión y visualización de datos.

Una vez obtenida y procesada la medida de CO, hay que encontrar una manera de mostrársela al usuario final.

Por diseño, el dispositivo no incluye ningún dispositivo de salida para mostrar información directamente, como podría ser una pantalla LCD o un monitor, ya que está pensado para poder ser colocado en lugares poco accesibles así como tener un bajo consumo eléctrico, cosa que no sería posible con los mencionados aparatos.

Por tanto, se precisa de alguna tecnología que sirva para transmitir datos de forma inalámbrica entre la placa y algún otro dispositivo, como podría ser un ordenador. De entre las muchas existentes se ha planteado el uso de tres: Bluetooth, LoRa y wifi.

4.2.1 LoRa

LoRa (Long Range) es una tecnología que utiliza un tipo de modulación en radiofrecuencia que sirve para transmitir datos a largas distancias con un consumo bajo de energía. Está basado en las técnicas de modulación de señal *spread spectrum* (amplio espectro).



Figura 12. Logo tecnología LoRa⁸

Esta tecnología se puede encontrar en cada vez más dispositivos IoT, especialmente aquellos diseñados para su uso en lugares donde hay poca cobertura de comunicaciones y es difícil conseguir una conexión eléctrica. Según la compañía que actualmente tiene la patente, Semtech Corp, puede llegar hasta 5 km en zonas urbanas y 15 km en campo abierto [14].

⁸ Extraída de: <https://www.thethingsnetwork.org/community/santa-rosa/post/que-es-la-tecnologia-lora-y-por-que-es-importante-para-iot>

Sin embargo, las técnicas de amplio espectro que emplea tienen el inconveniente de tener un ancho de banda más bajo que otras tecnologías inalámbricas por lo que no es lo más adecuado para sistemas que han de enviar mucha información en poco tiempo.

El estándar para la realización de la comunicación se conoce como LoRaWAN, que es el protocolo abierto de comunicación a Nivel de red [15]. Con él, se gestiona la comunicación de dispositivos que van sobre la capa física LoRa.

4.2.2 Wifi

La tecnología de comunicación inalámbrica wifi es de sobra conocida, ya que es ampliamente utilizada para transmitir datos y conectarse a Internet en hogares por todo el mundo mediante el uso de un router o punto de acceso de red.

La gran mayoría de dispositivos IoT que podemos encontrar actualmente en mercado utilizan esta tecnología por varios factores: lo fácil que es de integrar, requiriendo únicamente un enrutador, la posibilidad de conexión de varios dispositivos, que cuenta con un considerable alcance sin la necesidad de usar una configuración de tipo malla y que su velocidad de conexión y transmisión de información es bastante rápida, lo cual es ideal para aplicaciones que requieren enviar bits muy de seguido y un alto ancho de banda.

Uno de sus principales inconvenientes es su alto consumo energético, requiriendo además de una corriente estable y más alta que otros dispositivos, lo cual limita mucho su incorporación en aparatos alimentados por baterías o pilas

Esta tecnología está regida por el estándar del *Institute of Electrical and Electronics Engineers* (IEEE) 802.11n.

4.2.3 Bluetooth

El Bluetooth es un protocolo de comunicaciones diseñado para conectar dispositivos de bajo consumo por radiofrecuencia en la banda ISM (bandas de radio industriales, científicas y médicas). Cuenta con un alcance limitado, por lo que normalmente se utiliza para conectar dispositivos informáticos o de telecomunicaciones que se encuentran cerca los unos de los otros.

Es una conexión de fácil configuración y no requiere de un dispositivo externo para funcionar. Esto, sumado a que su gasto de energía es reducido, ha dado lugar a que en el mercado se puedan encontrar muchos dispositivos con compatibilidad bluetooth.



Figura 13. Logo Bluetooth ⁹

Como se ha comentado, su principal desventaja es su bajo rango de funcionamiento, no pudiendo normalmente utilizarse en aplicaciones que requieran la conexión de dispositivos que no se encuentren próximos. Esta distancia de conexión está determinada por la “clase” de bluetooth que utiliza el dispositivo, la cual determina su potencia de transmisión y cobertura [16].

	Alcance (m)	Potencia máxima (mW)
Clase 1	100	100
Clase 2	20	2.5
Clase 3	1	1
Clase 4	0,5	0.5

Tabla 3: Resumen de las características de cada clase de Bluetooth

La tecnología bluetooth ha sufrido varias revisiones, cada una con mayores prestaciones, encontrándose actualmente en el bluetooth 5.4.

4.2.4 Comparación y selección de tecnologías de transmisión datos.

Una vez consideradas algunas de las opciones para la conexión y transmisión de datos que se pueden encontrar actualmente en aparatos de similares características al del proyecto que nos ocupa, se va a realizar una comparación entre ellas para seleccionar la más indicada.

Es importante destacar que estas tecnologías pueden tener diferentes características según su versión y módulo que las incorpore, así que se han tomado los datos más genéricos pero que mejor se adapten a las necesidades funcionales del dispositivo.

⁹ Extraída de: <https://www.bluetooth.com/>

Propiedad	LoRa	Wifi 802.11 b/g/n	Bluetooth LE
Alcance Interior (exterior)	Muy alto 5 km (15 km)	Moderado 50 - 120 m (140 – 300 m)	Bajo 50 m
Velocidad de transmisión	0.3 Kbps - 50 Kbps	11 – 600 Mbps	1 Mbps
Consumo	Muy bajo	Alto	Bajo
Sencillez integración	Baja	Media - alta	Alta
Banda frecuencia	863 – 873 MHz	2.4 y 5 GHz	2.4 GHz

Tabla 4: Propiedades y comparación de las tecnologías inalámbricas seleccionadas.

La tabla 4 muestra algunas de las características ya mencionadas anteriormente, como el alcance y el consumo y, en ella, se resalta la mejor tecnología para cada propiedad.

Respecto al alcance, la técnica LoRa es la que mejores niveles proporciona mientras que el Bluetooth se queda demasiado corto en cuanto a rango, no siendo por tanto apto para el proyecto, a pesar de tener una sencillez de integración con otros dispositivos muy alta.

De las dos restantes ambas tienen sus pros y sus contras, siendo más apta para algunas condiciones como la distancia de transmisión de datos y el consumo la tecnología LoRa y en cuanto a velocidad de transmisión y ancho de banda el wifi (lo que se traduciría en mayor frecuencia de envío de bits). Este último factor, sin embargo, no se considera especialmente limitante considerando el uso para el que está diseñado el dispositivo de medición de CO₂, ya que, como muchos otros aparatos de domótica, no precisa enviar grandes tramas de datos ni de forma muy inmediata.

En la misma línea, el consumo energético suele ser un factor vital a la hora de diseñar un aparato IoT portable separado de la red eléctrica que funcione con baterías, sin embargo, no es este el caso, y por ello no se ha considerado un factor decisivo.

Por último, otro factor algo más subjetivo es el de la experiencia de usuario y facilidad de integrar el dispositivo en un entorno inalámbrico ya establecido. En este aspecto, la tecnología wifi es claramente ganadora, tanto por su uso más extendido que permitiría al usuario final entender y conectar con sencillez el aparato a su red privada como por la popularidad de su nombre y asociación con el auge de las tecnologías del *Internet of Things* y la domótica. En un hipotético caso de introducción del dispositivo al mercado, los términos wifi y Bluetooth tienen más fuerza por ser conocidos por personas alejadas del mundo de la electrónica.

En conclusión, hay diversos factores significativos para tener en cuenta a la hora de seleccionar cómo transmitir la información tomada por el sensor de un dispositivo a otro. Considerando varios de ellos, finalmente se ha optado por utilizar la tecnología wifi, principalmente por tener un rango de conexión que, aunque no es el mayor de todas las opciones, es suficiente para el uso que se precisa, cuenta con una gran facilidad a la hora de ser integrado y conectado con otros dispositivos que utilizan la misma tecnología, proporciona la capacidad de conexión a internet y es una de las opciones más populares en aparatos de similares características.

4.3 Control

Una parte muy importante del proyecto es la gestión de sus diferentes componentes y procesos. Es preciso ejecutar una serie de instrucciones para que el sistema al completo funcione, por lo que se necesita un elemento que controle todo lo que ocurre. En este caso se ha optado por el uso de un microcontrolador.

Un microcontrolador es un circuito integrado programable compuesto de una Unidad Central de Procesamiento (CPU), memoria y unidades de entrada y salida (E/S). Es capaz de ejecutar un programa almacenado en su memoria, por lo que se utiliza para controlar el funcionamiento de una determinada tarea. Tienen un tamaño reducido, de modo que es posible incorporarlos en el propio dispositivo que gobiernan, son fiables y generalmente baratos, por lo que son la base de los conocidos como sistemas embebidos, como es el caso de este proyecto.

Para facilitar el diseño y el montaje se optó por utilizar un kit de desarrollo de microcontroladores comercial. Los kits o placas de desarrollo son un método sencillo para generar un producto sin necesidad de fabricar una PCB (placa de circuito impreso) personalizada. Están formados por placas y componentes populares, así como conexiones para el cableado y la alimentación, de modo que el montaje sea simple y claro.

Hay abundantes modelos de este tipo de placas para una amplia gama de aplicaciones, algunos centrados en componentes específicos como acelerómetros, sensores, Bluetooth, wifi, ethernet, *displays* o altavoces, por ejemplo.

Además, hay diversas marcas que venden estos kits de desarrollo de microcontroladores, entre ellas Texas Instruments, MikroElektronika, STMicroelectronics, Microchip Technology Inc, Adafruit Industries, o, una de las más conocidas en el ámbito educativo, Arduino.

Para seleccionar la más adecuada para incluir en el proyecto, la placa de desarrollo escogida ha de ser económica, fácil de usar y de programar, así, de este modo, se reduce el

tiempo y el coste de desarrollo del producto. Como estas condiciones las cumplen varios kits de las marcas anteriormente mencionadas, se redujo el listado a las tres con las que más familiarizado se estaba: Arduino, ESP32 y STMicroelectronics.

Puesto que en el apartado anterior se ha determinado el uso de wifi como tecnología para la transmisión de datos, se han considerado algunas placas que incorporan chips wifi, pero no ha sido requisito de selección, puesto que estos chips se pueden utilizar de forma externa con módulos que los incorporen

4.3.1 Arduino

La marca Arduino hace referencia a 3 cosas: un lenguaje de programación libre similar a C++, un software libre multiplataforma que sirve para programar microcontroladores y, el que se va a desarrollar en este caso, una placa hardware libre que incorpora un microcontrolador programable y una serie de pines conectados a sus patillas de entrada/salida que facilitan la conexión de sensores y actuadores.



Figura 14. Logo Arduino ¹⁰

Los kits de desarrollo de Arduino son probablemente la opción más común para la iniciación en proyectos simples que necesitan un microcontrolador, ya que son fáciles de manejar y programar (sobre todo desde el IDE de Arduino) por eso son tan comunes en centros educativos y para aficionados con interés en la electrónica.

Hay varios tipos de placas Arduino, similares entre ellas, pero cada una con unas características específicas. Para este proyecto se va a estudiar la viabilidad de dos de ellas.

4.1.1.1 Arduino nano

Arduino Nano es una de las versiones más pequeñas de las placas de desarrollo de esta marca (18 x 45 mm), siendo por tanto una de las más convenientes por tamaño.

¹⁰ Extraída de: <https://es.wikipedia.org/wiki/Arduino>

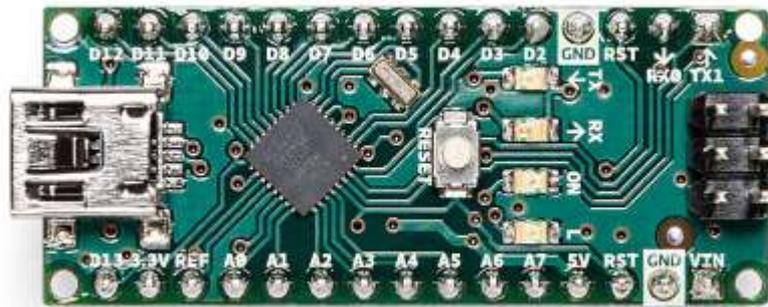


Figura 15. Alternativa 1 de control: Arduino nano ¹¹

De entre las características del dispositivo, las más relevantes para el proyecto son su microcontrolador ATmega328 (8 bits), su voltaje de funcionamiento de 5 V y su frecuencia de reloj de 16 MHz. Además, este microcontrolador cuenta con numerosos pines, tanto digitales como analógicos, lo que le da gran versatilidad a la placa en diferentes aplicaciones.

Entre otras propiedades del dispositivo, se destacan:

- Memoria: Cuenta con varias memorias para diferentes usos. Por un lado, para almacenar el programa que se quiere ejecutar, contiene una memoria flash de 32 KB, con 2 KB utilizados en el sistema de arranque. Por otro, cuenta con 1 KB de memoria EEPROM y 2 KB de SRAM.
- Comunicación: Para transmitir el programa desde entorno de desarrollo al microcontrolador o para imprimir datos por el puerto serie dispone de una conexión mediante puerto micro USB.
- Pines: Como muestra la Figura 16, esta placa tiene un número considerable de pines. Entre ellos, se distinguen 14 pines digitales (de entrada o salida). Además, 6 de estos en concreto se pueden aprovechar como salidas PWM. Por otra parte, cuenta también con 6 pines de entrada analógicos, útiles en este caso para leer valores de sensores.

¹¹ Extraída de: <https://store.arduino.cc/products/arduino-nano>

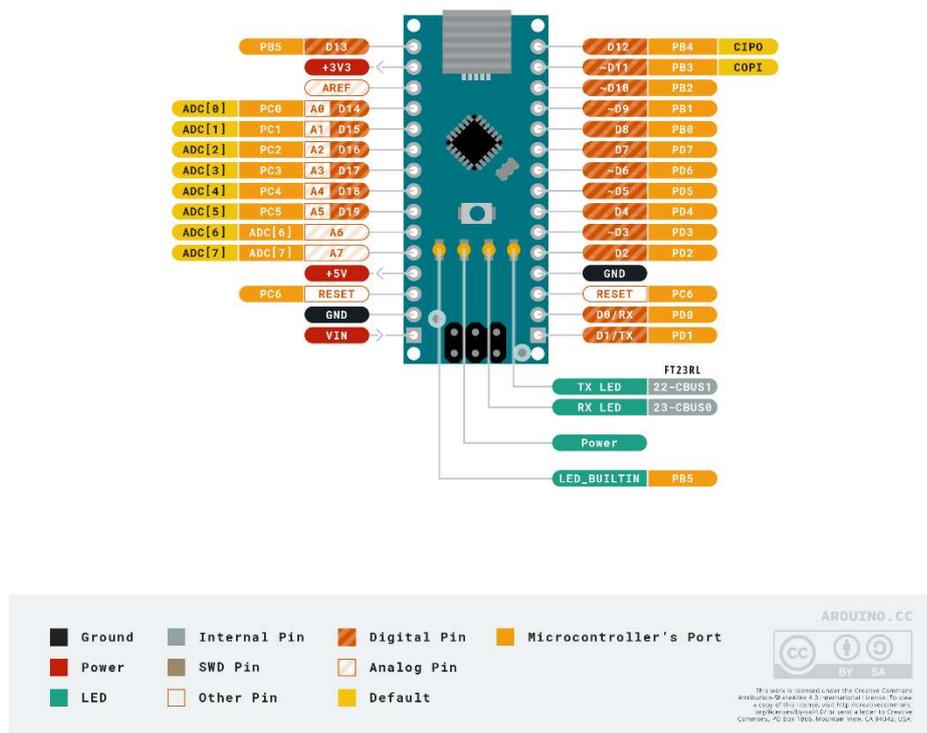


Figura 16. Distribución de pines del Arduino nano ¹²

En cuanto a la conectividad, el Arduino nano original no incorpora wifi ni Bluetooth, pero sí hay disponibles diferentes versiones en el mercado con otras características técnicas que incorporan este tipo de conexiones, entre ellas el Arduino nano 33 IOT, el Arduino nano 33 BLE sense y el Arduino nano RP2040, que son opciones viables en caso de querer utilizar una placa de esta marca y modelo.

4.1.1.2 Arduino Uno R4 WiFi

Arduino UNO es la versión más estándar de la placa Arduino, actualmente en su cuarta revisión desde que salió al mercado en 2010.

Esta versión, de 32 bits, combina el microprocesador RA4M1 de Renesas Technology Corporation con el ESP32-S3 de Espressif Systems. Tiene wifi y Bluetooth incorporados, de modo que está particularmente planteado para el uso en dispositivos IoT y de domótica del hogar. Dispone además de una matriz de leds de 12x8 incorporada, que se puede utilizar para mostrar símbolos o mensajes básicos al usuario.

¹² Extraída de: <https://docs.arduino.cc/hardware/nano>

- Memoria: cada uno de los microprocesadores cuenta con su propia memoria. El RA4M1 tiene 32 KB de RAM y 256 KB de memoria flash y el ESP32-S3 384 KB de ROM (memoria de solo lectura) y 512 KB de SRAM.
- Comunicación: Para transmitir el programa a la placa se utiliza la conexión USB tipo C. Además, cuenta con interfaces UART, 1 I2C, 1 SPI y *Controller Area Network* (CAN) para conectarse con otros dispositivos, además de las ya mencionadas conectividades wifi y Bluetooth.
- Alimentación: la placa al conjunto requiere de 5 V para funcionar, el ESP32 de 3,3 V.
- Pines: consta de 14 puertos de entrada/salida digital, 6 analógicos y pines dedicados a las conexiones tipo I2C, SPI y UART.

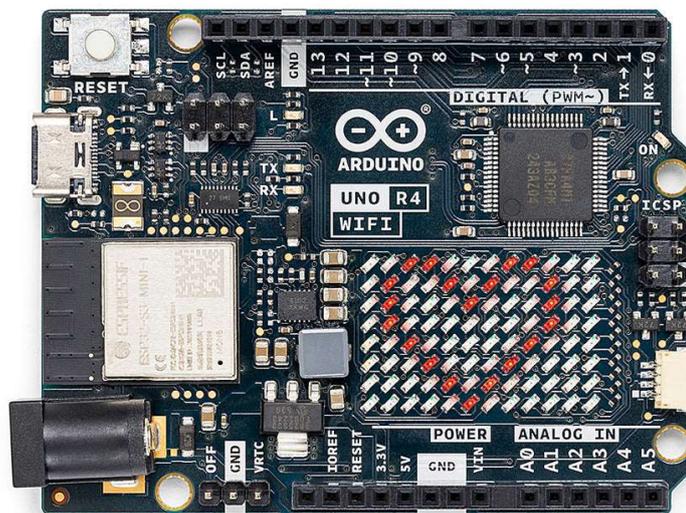


Figura 17. Alternativa 2 de control: Arduino Uno R4 WiFi ¹³

Una de las ventajas de esta placa es su compatibilidad de hardware con muchos componentes distintos, incluidos los de la versión anterior, ya que mantiene el mismo factor de forma, disposición de los pines (*pinout*) y voltaje de operación que el UNO-R3, placa para la cual hay incontables accesorios.

4.3.2 ESP32

ESP32 es una familia de chips de bajo coste desarrollada por la empresa china Espressif Systems. Utiliza un microprocesador Tensilica Xtensa LX6, que trabaja con frecuencias de

¹³ Extraída de: <https://store.arduino.cc/products/uno-r4-wifi>

hasta 240 MHz, y es particularmente popular debido a que integra tecnología wifi y Bluetooth. Además, está especialmente diseñado para ser usado en dispositivos móviles y aplicaciones de IoT, sobre todo a causa de su bajo consumo de energía y diferentes modos de potencia.

Puede funcionar por sí mismo controlando un dispositivo o junto a otro microcontrolador, reduciendo carga del procesador principal y otorgando al sistema funcionalidades wifi y Bluetooth mediante interfaces SPI /SDIO o I2C / UART.

Igual que Arduino cuenta con varios módulos y kits de desarrollo apropiados para diferentes aplicaciones.

4.1.1.3 ESP32-DevKitC-32E

El ESP32-DevKitC-32E es un kit de desarrollo que incluye el módulo ESP32 ESP-WROOM-32E, basado en el chip ESP32 y muy utilizado en aplicaciones relacionadas con Internet of Things (IoT), sensorización y automatización. Entre las especificaciones más importantes de este kit de desarrollo destacan:

- Memoria: tiene una SRAM de 520 KB y una memoria flash de 4 MB.
- Comunicación: Para transmitir el programa utiliza la conexión micro USB. Por otra parte, dispone de una variedad de interfaces para la comunicación con otros dispositivos, incluyendo 3 interfaces UART, 2 I2C, 3 SPI y *Controller Area Network* (CAN). Además, y una de las razones por la que los módulos que incluyen el chip ESP32 son tan populares, cuenta con capacidad de conexión inalámbrica a otros dispositivos mediante redes wifi y bluetooth.
- Alimentación: requiere de 5 V para funcionar. Estos pueden ser suministrados mediante una conexión al USB de un ordenador o con una fuente de alimentación estándar.
- Pines: Incluye 25 pines GPIO, de entre los cuales 4 son únicamente de entrada y 16 pueden usarse como canales de salida PWM. Además, tiene 18 canales de entrada analógica y un conversor analógico-digital con una resolución que se puede ajustar entre 9 y 12 bits.

Además, este módulo puede ser programado con el lenguaje Arduino y es compatible con el entorno de desarrollo Arduino IDE, en el cual se han incorporado varias librerías de para los ESP32, lo que facilita enormemente el proceso.



Figura 18. Alternativa 3 de control: ESP32-DevKitC-32E ¹⁴

4.3.3 STM32

STM32 es una familia de microcontroladores de 32 bits basados en los procesadores Arm Cortex-M. Fue desarrollada por la por la compañía francoitaliana STMicroelectronics, uno de los mayores exponentes en el mundo de la producción de semiconductores, chips, periféricos informáticos y otros elementos electrónicos.



Figura 19. Logo ST ¹⁵

Estos microcontroladores están caracterizados por su alta potencia de procesamiento, conectividad y su bajo consumo de energía y son una opción muy común para aplicaciones de sistemas embebidos, especialmente por trabajar con los estándares de la industria.

¹⁴ Extraída de: <https://www.digikey.es/es/products/detail/espressif-systems/ESP32-DEVKITC-32E/12091810>

¹⁵ Extraída de: https://en.wikipedia.org/wiki/File:STMicroelectronics_2020_logo.svg

Además, la empresa comercializa gran cantidad de herramientas de software, entre ellos un entorno de desarrollo propio, y variedad de periféricos adaptables, por lo que también son una buena opción para pequeños proyectos y soluciones automáticas integrales [17].

Dentro de esta familia hay bastante diversidad en cuanto a capacidades y modelos. Para entenderlos, primero hay que echar un vistazo a los procesadores Arm Cortex-M.

Los ARM Cortex-M están desarrollados por la empresa inglesa Advanced RISC Machines Holdings Limited (ARM). ARM se dedica principalmente al diseño y licencia de procesadores, chips y herramientas de software para trabajar con ellos (como DS-5 o Keil MDK). A partir del trabajo de ARM, otras empresas luego compran las licencias del diseño en forma de *Verilog Intellectual Property* (IP) y distintas herramientas de la compañía y así desarrollan y fabrican los procesadores ARM, complementando y adaptando lo que consideren. Entre ellas podemos encontrar, por ejemplo, a Texas Instruments, ST, Microchip, Silicon Lab o Infineon.

Los procesadores ARM se pueden encontrar en dispositivos de todos los tipos, desde ordenadores hasta placas baratas, ya que existen varias familias y modelos de estos chips. Actualmente la empresa utiliza la denominación “Cortex” seguido de una letra para distinguir las aplicaciones para las que están diseñados sus procesadores:

- Cortex-A (*Application*): Para aplicaciones de alto rendimiento, como móviles, tablets, televisiones y mini PCs. Son procesadores capaces de ejecutar un sistema operativo y se utilizan en placas como las Raspberry Pi, Orange Pi y BeagleBone.
- Cortex-R (*Real Time*): Para aplicaciones de tiempo real, sobre todo en dispositivos de redes de datos, en automóviles y en robots.
- Cortex-M (*Microcontroller*): Usados en aplicaciones sencillas, de bajo coste y consumo de energía, especialmente en sistemas embebidos.
- Cortex-X (*Custom*): Aplicaciones muy exigentes en cuanto a potencia de procesamiento, como ordenadores. El rendimiento es lo más importante.

Considerando el proyecto a mano, los procesadores ARM Cortex-M son los más adecuados para cumplir con las especificaciones técnicas.

Sin embargo, dentro de esta familia de procesadores también hay distintas variantes con diferentes características y los microcontroladores STM32 cuentan con modelos con cada una de ellas. Entre algunos de los diferentes ARM Cortex-M, podemos encontrar el M0, el más simple y pequeño de todos, pensado sobre todo para dispositivos en los que prima el bajo precio y consumo de energía, y el M0+, una versión mejorada, con mayor potencia y menor consumo. También el M3 y el M4, ambos de alto rendimiento y aptos para realizar

tareas complejas, pero a costa de mayor superficie de silicio para fabricar (y por tanto más caros) y el M7, similar a estos, pero con mejor gestión de la energía y con buses internos de datos de 64 bits en vez de 32 [18].

Teniendo en cuenta las particularidades de los procesadores en los que están basados, las diferentes versiones de STM32 se entienden mejor. Hay algunas más centradas en la potencia y las altas prestaciones, que suelen incluir los Cortex M3, M4 y M7, otras en su uso para aplicaciones comunes de microcontroladores, las centradas en el bajo consumo energético para funcionar sobre todo junto a baterías y las que tienen algún tipo de conectividad inalámbrica.

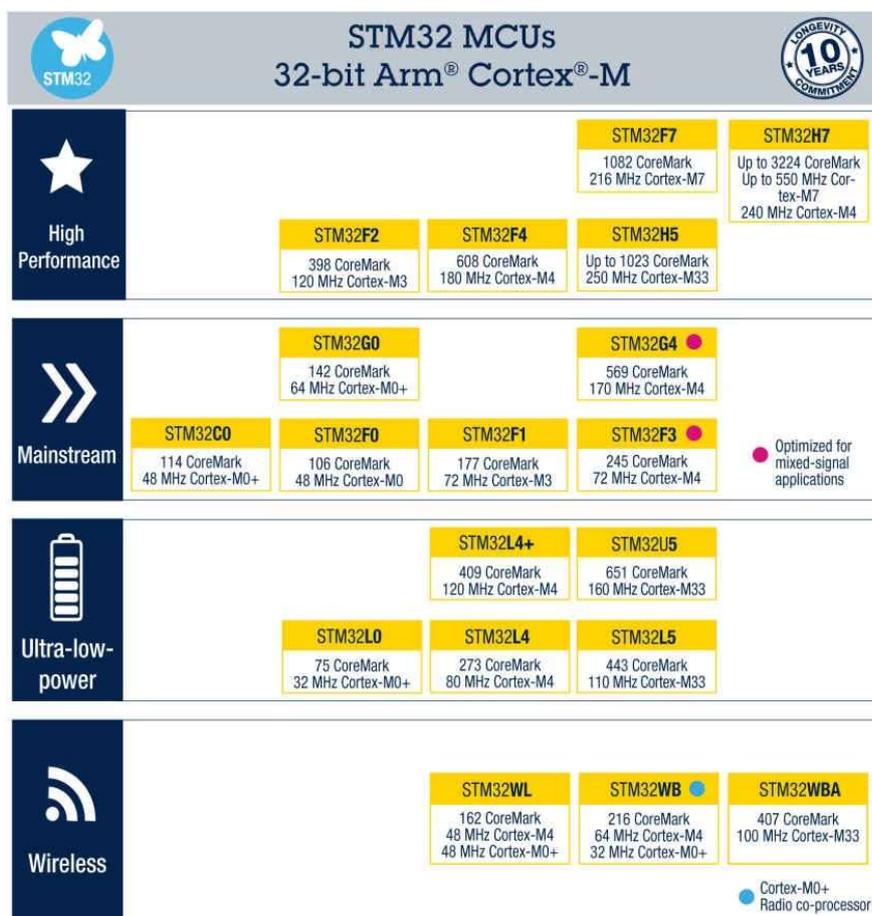


Figura 20. STM32 32-bit Arm Cortex-M MCUs ¹⁶

4.3.3.1 STM32L053R8

De entre las muchas opciones de microcontroladores que dispone la gama STM32, aquí se va a profundizar en el STM32L0.

¹⁶ Extraída de: <https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html>

Este microcontrolador está diseñado para funcionar con poca potencia, ya que está basado en el ARM Cortex M0+ de 32 bits, uno de los procesadores de la marca con mejor gestión energética. Funciona a un máximo de 32 MHz de frecuencia de reloj y dispone de varias memorias: 64 KB de memoria Flash, 8 Kb de RAM y una EEPROM de 2 KB para almacenamiento permanente de datos. Además, cuenta con distintos modos de energía para ajustar el consumo.

Para poder trabajar cómodamente sus microcontroladores, ST ofrece la posibilidad de comprar distintas placas de desarrollo que los incluyen, las conocidas como placas STM32 núcleo.

Estas placas, o boards, en inglés, son muy populares debido a su facilidad de uso y bajo precio y están pensadas para que la mayor parte de la gente pueda realizar proyectos que requieran de control con ellas. Las hay de tres tipos: las Nucleo-32, las Nucleo-64 y las Nucleo-144, según el número de pines con los que cuentan.



Figura 21. Comparativa boards STM32 ¹⁷

Las boards Nucleo vienen integradas con un *debugger* STLINK, que permite el depurado de programas sin necesidad de recurrir a sondas externas. Además, están diseñadas para usar los mismos conectores que las placas de la marca Arduino por lo que muchos de los periféricos diseñados para ellas pueden ser aprovechados.

Existen varias placas que integran el STM32L0 como la Nucleo-L073RZ, la Nucleo-L011K4 o la Nucleo-L010RB, entre otras. En este caso, se ha escogido como opción la Nucleo-L053R8, cuyas características son las siguientes:

¹⁷ Extraída de: <https://www.st.com/en/evaluation-tools/stm32-nucleo-boards.html>

- Comunicación: Para conectarse a la board y enviar el programa se utiliza la conexión micro USB. Además, cuenta con 2 interfaces USART, 1 UART, 2 I2C, 4 SPI de 16 Mb/s.
- Alimentación: requiere de una fuente de 1.65 V a 3.6 V para su operación. Asimismo, cuenta con varios modos de funcionamiento, espera y Standby.
- Pines: cuenta con 51 pines GPIO y la mayoría pueden ser programados como digitales o analógicos. Para conectarse a ellos cómodamente, la placa cuenta con conexiones externas macho y hembra compatibles con conectores Arduino. Asimismo, tiene disponibles un convertidor analógico-digital y otro digital-analógico, ambos de 12 bits.
- Cuenta además con un LED verde programable, un botón azul *user* también programable y un botón de *reset* negro.

Al seleccionar un microcontrolador apropiado para un proyecto hay varios factores importantes a considerar, entre ellos la arquitectura, la memoria disponible, el consumo energético, el coste monetario, los periféricos con los que se puede conectar y los protocolos con los que se puede comunicar (como USB, SPI, UART, I2C, etc.). Revisando lo que ofrece este STM32L0, sobre todo sus modos de baja potencia, y considerando las necesidades operativas del proyecto, es una opción muy adecuada. Esta razón, sumada a la familiaridad previa con las placas de desarrollo de STM32, ha llevado a la elección de la placa de desarrollo STM32L053R8 para desarrollar este trabajo.

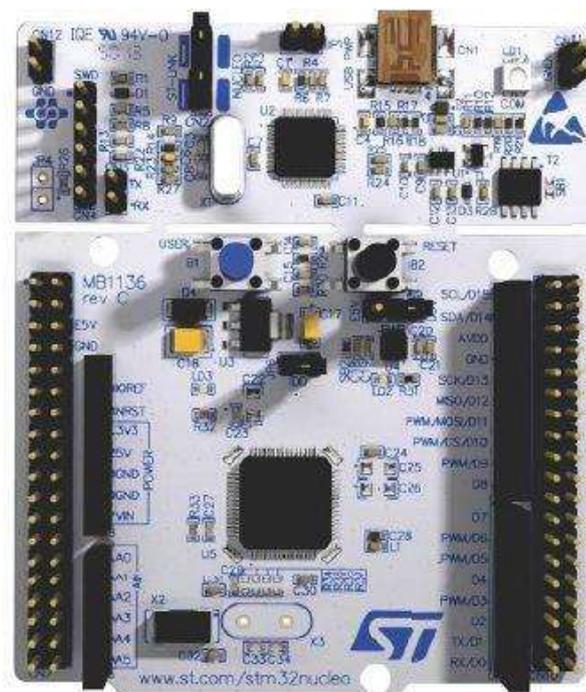


Figura 22. Alternativa 4 de control: Placa STM32L053R8 ¹⁸

4.4 Medidas de seguridad ante la presencia de CO.

Como ya se ha mencionado anteriormente, ciertas concentraciones de monóxido de carbono pueden llegar a ser perjudiciales para la salud, con síntomas desde dolores de cabeza, náuseas y vómitos, hasta insuficiencias cardíacas, dificultades para respirar, convulsiones, caídas en coma y, en los casos más graves, incluso la muerte. En espacios cerrados, la acumulación de este gas es mayor, por lo que su peligro se dispara.

Por esta razón, hay que cuidar que los lugares donde se emite CO en mayores cantidades, como en habitaciones con hornos y estufas, aparcamientos de coches y en industrias en las que se quema gas o carbón, cuenten con las medidas de protección necesarias. Estas están constituidas, normalmente, por 3 elementos: detección del nivel de monóxido de carbono, alarma en caso de superar el nivel seguro y, opcionalmente, un sistema de extracción de aire para sustituirlo por uno no contaminado. Puesto que el primero de ellos se discute en un punto anterior, en este apartado se estudiará la posibilidad de elaborar una maqueta que simule un sistema de alarma y ventilación.

¹⁸ Extraída de: <https://www.mouser.es/ProductDetail/STMicroelectronics/NUCLEO-L053R8?qs=fk8d1pkAUMtaeo4tSsfchQ%3D%3D>

4.4.1 Extracción de aire

Comenzando con la ventilación, como ya se ha comentado, se va a plantear la realización de una maqueta cuyo funcionamiento simule un sistema de extracción de aire real, de modo que, idealmente, se pudiera reutilizar la etapa de control con ligeras modificaciones.

Para renovar el aire en espacios interiores y garantizar su salubridad, una opción muy común es utilizar ventiladores accionados por motores eléctricos, los cuales impulsan el aire contaminado e introducen al mismo tiempo aire limpio y renovado fresco por depresión.

4.4.1.1 Control

Para controlar este tipo de motores, se pueden aplicar varios métodos. En este caso se va a profundizar en el control todo/nada y la regulación por ancho de pulso.

4.4.1.1.1 Control todo/nada

El control todo/nada (o control ON/OFF) es una de las formas más sencillas de gobernar un sistema. En este caso, consistiría en activar el ventilador y ponerlo a máxima potencia cuando la medición de CO en el aire supere un valor límite y apagarlo en cuando este valor sea inferior a dicho umbral.

Este es uno de los controles más sencillos de implementar, pero tiene considerables limitaciones y una respuesta bastante pobre. Su principal inconveniente en este caso sería la repetitividad de la conmutación: al llegar al valor de CO asignado, el ventilador se encendería, disminuyendo rápidamente la concentración de gas y, en consecuencia, apagándose en el siguiente ciclo de lectura del sensor. Si el CO siguiera aumentando, el motor se activaría y desactivaría cada vez que se llegara al límite establecido, lo cual podría acabar dañándolo.

Una solución a este problema es aplicar el llamado control todo/nada con histéresis, que consiste en fijar un valor umbral para la activación y otro distinto para la desactivación. De este modo, el ventilador no tendría que conmutar muchas veces y podría llegar a estabilizarse al cambiar de un estado a otro.

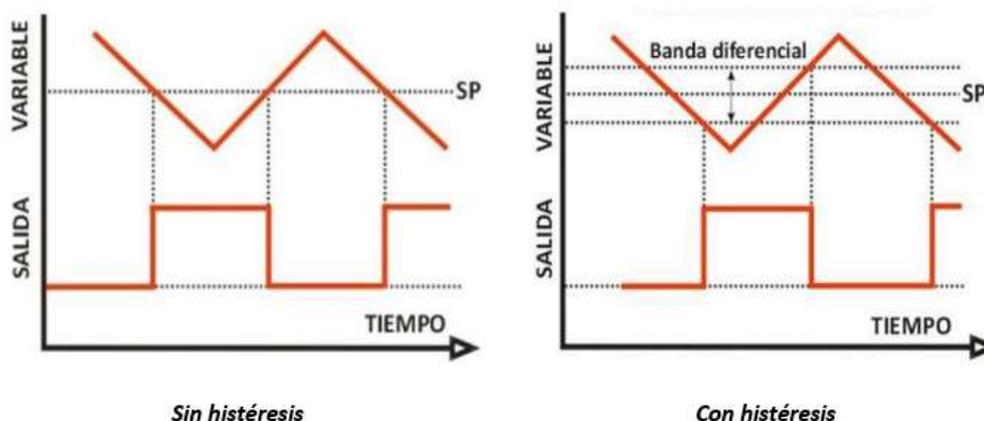


Figura 23. Control todo/nada sin y con histéresis ¹⁹

4.4.1.1.2 Modulación por ancho de pulso

Un método muy común para controlar un motor eléctrico que hace girar un ventilador es el de la modulación por ancho de pulsos, de modo que le permita extraer distintos caudales de aire en cada momento.

La modulación por ancho de pulso es una técnica que se utiliza para crear una señal analógica a partir de una fuente digital. Consiste en generar una señal digital y variar el tiempo que se encuentra en estado alto o bajo, ajustando el ancho de pulso en función de ese tiempo. El motor finalmente recibe la tensión media de la señal, que se hace más alta a medida que aumenta el ancho de pulso de la señal PWM.

Una señal PWM tiene dos componentes principales: el ciclo de trabajo y la frecuencia. El ciclo de trabajo es una medida del tiempo que la señal PWM está encendida con respecto al periodo de ciclo completo y se puede modificar para regular la cantidad de energía que le llega al motor. La frecuencia, por su parte, es una medida de lo rápido que la señal PWM alterna entre los estados alto y bajo y se mide en hercios.

¹⁹ Extraída de: <https://www.domoticadomestica.com/termostatos-inteligentes-control-pid/>

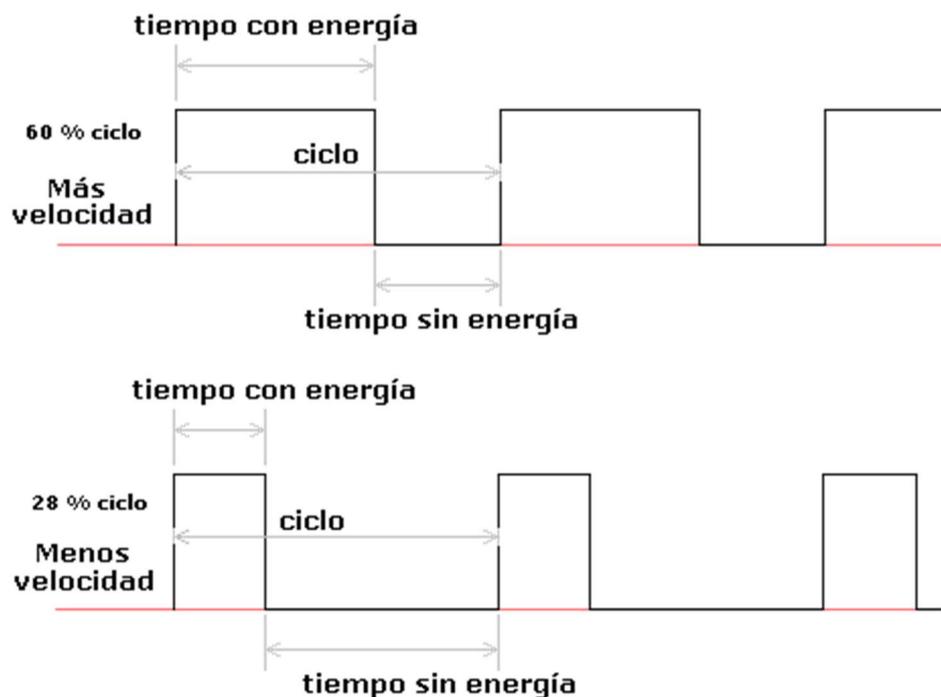


Figura 24. Dos señales PWM con ciclos de trabajo distintos ²⁰

En la Figura 24 se puede ver una señal PWM con dos ciclos de trabajo diferentes. El ciclo de trabajo es el ancho de la parte positiva de la señal y se expresa en forma de porcentaje respecto del periodo de la señal. A mayor ciclo de trabajo, mayor cantidad de energía recibe el dispositivo que se está controlando.

Si se comparan las dos opciones, el control todo/nada tiene el inconveniente de no poder regular la potencia, solo encender o apagar el motor. Esto provocaría que, si el valor de monóxido de carbono oscilara alrededor de su punto de activación, el motor se encontraría en constante estado de conmutación en su máxima potencia, lo que le causaría un deterioro más rápido.

Por el contrario, el control por modulación de ancho de pulso permite hacer girar el ventilador a distintas velocidades dependiendo del ancho de banda, lo que se traduciría en mayor velocidad a mayor concentración de CO. Al no estar siempre al máximo de sus prestaciones, la vida útil sería más larga. Por esta razón, se ha elegido este tipo de control para el motor.

²⁰ Extraída de: <https://robots-argentina.com.ar/didactica/control-de-motores-de-cc-por-ancho-de-pulso-pwm/>

4.4.1.2 Ventilador real. Características.

Una vez seleccionado el método para controlarlo, es preciso encontrar un ventilador eléctrico que pueda ser controlado por PWM. Hay incontables opciones en el mercado, muchas de ellas válidas. En este caso se ha seleccionado el W3G300-UL11-35, de la marca Ebm-papst [19].



Figura 25. Ventilador axial Ebm-papst W3G300-UL11-35 ²¹

El W3G300-UL11-35 es un ventilador axial de 3,2 kg de peso y 400 mm de diámetro, diseñado para su uso únicamente en sistemas estacionarios. Utiliza un motor eléctrico M3G055-DF monofásico, de 230 V de voltaje nominal a una frecuencia de 50/60 Hz. Tiene una potencia de 120 W y una velocidad de giro máxima de 1800 revoluciones por minuto.

La característica que más interesa para este proyecto es la posibilidad de controlarlo por modulación de ancho de pulso. En concreto, en su *datasheet* [19] se especifica que puede ser controlado mediante una señal DC de 0 a 10 V PWM (Figura 26).

²¹ Extraída de: <https://www.ebmpapst.com/at/en/products/axial-fans/p/W3G300UL1135.html>

Technical features	<ul style="list-style-type: none"> - Output 10 VDC, max. 10 mA - Tach output - Power limiter - Motor current limitation - Soft start - Control input 0-10 VDC / PWM - Control interface with SELV potential safely disconnected from the mains - Overvoltage detection - Thermal overload protection for electronics/motor - Line undervoltage detection
---------------------------	--

Figura 26. Características técnicas ventilador W3G300-UL11-35

4.4.1.3 Maqueta control ventilador.

Habiendo seleccionado un ventilador ya existente, se plantea la posibilidad de simular su control mediante una maqueta. Para ello se van a utilizar tres componentes: un motor eléctrico, un módulo para el control PWM y un *encoder*.

4.4.1.3.1 Motor

En las principales páginas de venta componentes electrónicos podemos encontrar motores de pequeño tamaño válidos para el desarrollo de la maqueta.

La primera decisión que se debe tomar es con respecto a su alimentación: mediante corriente continua o alterna. Si bien el ventilador real utiliza un motor AC, estos son significativamente más caros y difíciles de utilizar. El objetivo principal de la maqueta es probar el control, por lo que la alimentación eléctrica se puede considerar secundaria y, puesto que un factor decisivo es la simplicidad en el desarrollo y los motores DC son significativamente más sencillos de manejar, se ha optado por este tipo.

En segundo lugar, dentro del grupo de motores de corriente continua existen dos tipos: con y sin escobillas. Se ha de escoger cuál se va a utilizar.

Los motores con escobillas (*brushed*) funcionan deslizando unas escobillas sobre un conmutador, creando así un contacto eléctrico que permite el paso de corriente y genera el movimiento. Aunque son una tecnología más antigua, por su bajo coste y fácil mantenimiento siguen siendo comunes. Sin embargo, tienen una menor eficiencia y vida útil que los motores sin escobillas debido principalmente a la fricción y desgaste causados por el contacto entre las escobillas y el conmutador.

Por otro lado, los motores sin escobillas (*brushless*), son más modernos y no utilizan conmutadores ni escobillas, la conmutación de las bobinas se realiza electrónicamente. Esto reduce la fricción, mejora la eficiencia energética, la vida útil y permite un giro a mayor velocidad.

Por tanto, se puede comprobar que ambas opciones tienen sus ventajas e inconvenientes. Puesto que para la maqueta no se precisa del mismo tipo exacto de motor que el ventilador real y los motores con escobillas son generalmente más baratos, se utilizará uno de este tipo.

Teniendo en cuenta las alternativas disponibles, se ha optado finalmente por el motor Nidec serie DMN29, de 12 V DC, 3 W de potencia y una velocidad de giro máxima de 3700 rpm [20].



Figura 27. Motor DC Nidec serie DMN29, 12 V²²

4.4.1.3.2 Módulo controlador

Normalmente, los microcontroladores funcionan con lógica TTL de 3,3 V a 5 V y corrientes menores a los 25 mA, mientras que los motores de corriente continua suelen precisar de una tensión e intensidad mayores para comenzar a girar. Por esta razón, es preciso utilizar un circuito de alimentación externo para proporcionar la energía suficiente al motor y un *driver* para controlarlo, es decir, un sistema de amplificación de corriente que reciba del microcontrolador una señal de control de baja intensidad y la convierta en una señal de corriente más alta con la que se pueda conducir el motor.

Puesto que el motor seleccionado es de corriente continua con escobillas y se ha determinado que se va a utilizar una placa de desarrollo con un microcontrolador STM32, la

²² Extraída de: <https://es.rs-online.com/web/p/motores-dc/8814475>

opción más viable para manejar el motor es utilizar un módulo de expansión diseñado para este tipo de motores y placas.



Figura 28. Placas de expansión STM32 Nucleo para el control de motores ²³

ST vende módulos para motores DC de varios tipos: paso a paso, con y sin escobillas, así como varios modelos con diferentes características para cada uno de los tipos, como capacidad para mayor tensión de alimentación, muy bajo consumo o control paralelo de dos motores.

²³ Extraída de:

https://www.st.com/resource/en/brochure/stm32_nucleo_expansion_boards_motor_control_selection_guide.pdf

Board	Voltage	Power/Max Current	ST parts	Stackable	Application focus
X-NUCLEO-IHM04A1	8 - 50 V _{DC}	Up to 2.8 A _{rms} (5.6 A _{rms} in parallel mode)	1x L6206	No	Stage lighting, antenna control, vending machines, factory automation. Both bi-directional or uni-directional brush DC motors without current limiting. It drives dual bipolar DC or quad unipolar DC motors. Relays and other inductive loads. Outputs can be paralleled.
X-NUCLEO-IHM12A1	1.8 - 10 V _{DC}	Up to 1.3 A _{rms}	1x STSPIN240	No	eValves, IoT, edu/home robots, healthcare, toys, eLock, actuators. Ultra low voltage and portable equipment. Dual bi-directional brush DC motor with current limiter. Parallel operation is not allowed.
X-NUCLEO-IHM13A1	1.8 - 10 V _{DC}	Up to 2.6 A _{rms}	1x STSPIN250	No	eValves, IoT, edu/home robots, healthcare, toys, eLock, actuators. Ultra low voltage and portable equipment. Single high current bi-directional brush DC motor with current limiter.
X-NUCLEO-IHM15A1	7 - 45 V _{DC}	Up to 1.5 A _{rms} (3 A _{rms} in parallel mode)	1x STSPIN840	No	Stage lighting, Industrial automation, service robots, medical and health care, ATM, Vending machines. Dual bi-directional brush DC motor with current limiter. Parallel operation enabled through a dedicated pin.

Figura 29. Placas de expansión STM32 Nucleo para el control de motores DC con escobillas y sus características ²⁴

Teniendo en cuenta el motor y el microcontrolador seleccionados y las opciones para módulos controladores, la opción más acorde es la placa X-NUCLEO-IHM13A1 para microcontroladores STM32-Nucleo, construida para manejar un motor DC con escobillas.



Figura 30. Alternativa 1 para el control de motores: Placa X-NUCLEO-IHM13A1²⁵

²⁴ Extraída de:

https://www.st.com/resource/en/brochure/stm32_nucleo_expansion_boards_motor_control_selection_guide.pdf

²⁵ Extraída de: <https://www.st.com/en/ecosystems/x-nucleo-ihm13a1.html>

4.4.1.3.3 Encoder

Un *encoder*, o codificador rotatorio, es un aparato electromecánico que se utiliza para medir la posición angular de un eje y convertirla a una señal digital. Se utilizan en gran variedad de aplicaciones, entre ellas como sensor de velocidad de giro y posición de motores.

En el caso de este proyecto sería interesante tener la capacidad de saber sin estar delante del motor si se encuentra girando y a qué velocidad lo está haciendo, de modo que se pudiera ser consciente de cuando se activa porque el ambiente podría estar contaminado o, en un hipotético caso de avería, se pudiera solucionar rápidamente. En definitiva, se quiere contar con un aparato que haga las veces de tacómetro digital.

Por ende, se va a buscar un *encoder* que sea tanto compatible con el motor seleccionado en cuanto a tamaño y velocidad de giro como con el microcontrolador seleccionado, para poder tomar la señal que genere en una de sus entradas.

Un requisito indispensable es que dicho codificador sea compatible en tamaño con el motor escogido, sobre todo con lo que respecta a la longitud y diámetro del eje. En la Figura 31, extraída del *datasheet* del motor [20], se han señalado estas características.

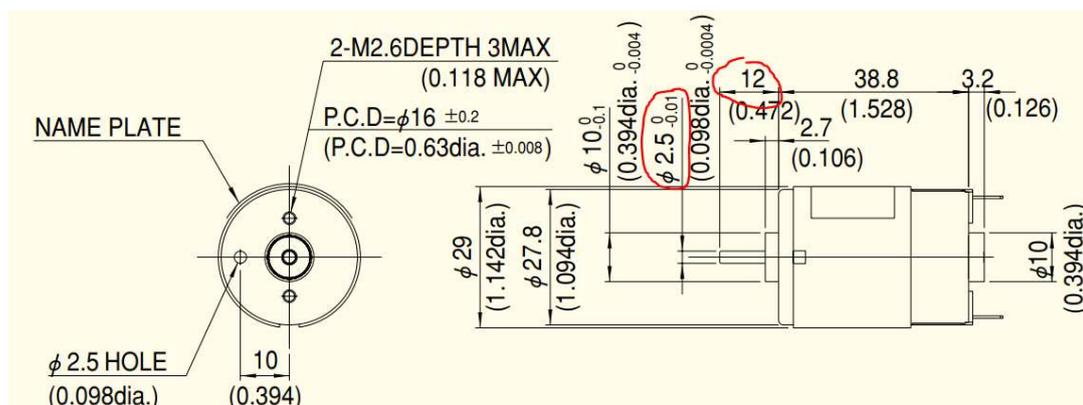


Figura 31. Plano motor DMN29

Asimismo, es importante que sea sencillo de instalar, programar y tenga un precio razonable. Considerando estas condiciones, y que se va a utilizar para medir únicamente la velocidad, una opción adecuada es utilizar un *encoder* incremental de la serie AMT10 de la marca CUI Devices, adecuado para ejes de mínimo 9 mm de longitud y 2 de diámetro.



Figura 32. Codificadores rotatorios AMT102 y AMT103 de CUI Devices ²⁶

Una ventaja de los codificadores incrementales es que son bastante económicos comparándolos con los ópticos y los absolutos, además de tener una buena precisión al medir.

Respecto a los distintos modelos dentro de la serie, el AMT102 tiene orientación radial y el AMT103 axial. Esto se traduce en una posición distinta de los pines con respecto al eje de giro: perpendicular y paralelo, respectivamente. Se escogerá el AMT103 para que los cables se conecten desde la parte frontal del motor.

²⁶ Extraída de: <https://www.cuidevices.com/product/resource/amt10-v.pdf>

5 DESCRIPCIÓN DE LA SOLUCIÓN ADOPTADA

En este apartado se detallan las características de los componentes elegidos de entre las distintas opciones para la creación del dispositivo y sus principios de operación.

El funcionamiento básico del producto consiste en la toma de medidas de monóxido de carbono y temperatura alrededor del sensor, su visualización a través de una interfaz abierta en un PC y la activación de una respuesta en caso de que el valor sobrepase un nivel perjudicial para los seres humanos. Este queda detallado en el siguiente diagrama de flujo (Figura 33), que se repetiría cíclicamente mientras el dispositivo esté en funcionamiento:

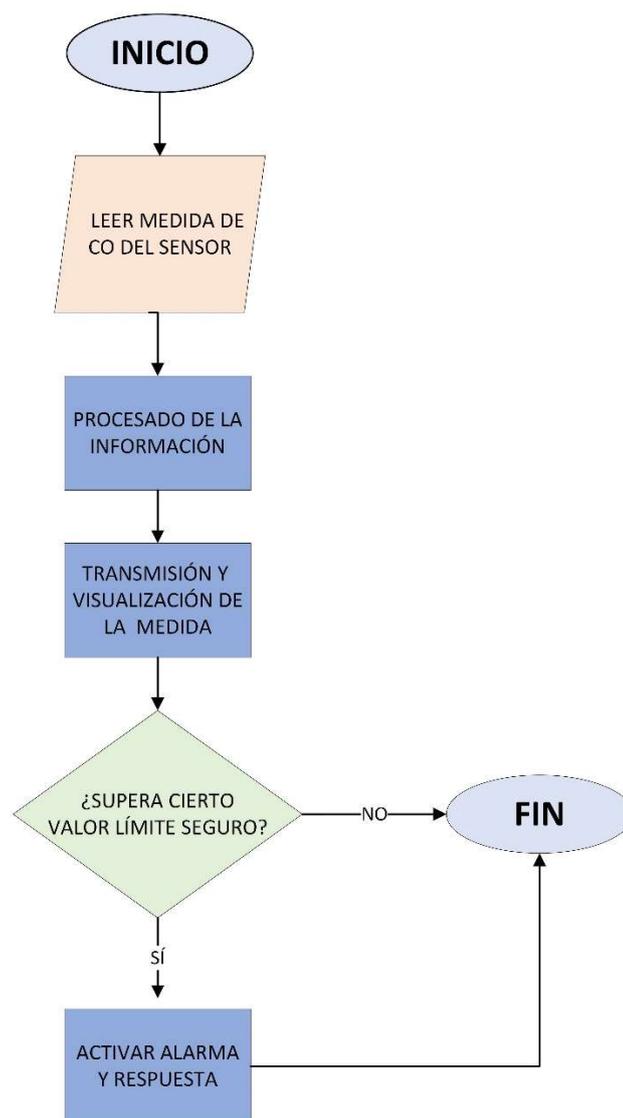


Figura 33. Diagrama de flujo funcionamiento del dispositivo (elaboración propia)

La Figura 34 muestra la estructura desglosada de este proyecto, incluyendo los diferentes subsistemas de los que está compuesto, así como los conjuntos y componentes que se incluyen en cada uno de ellos.

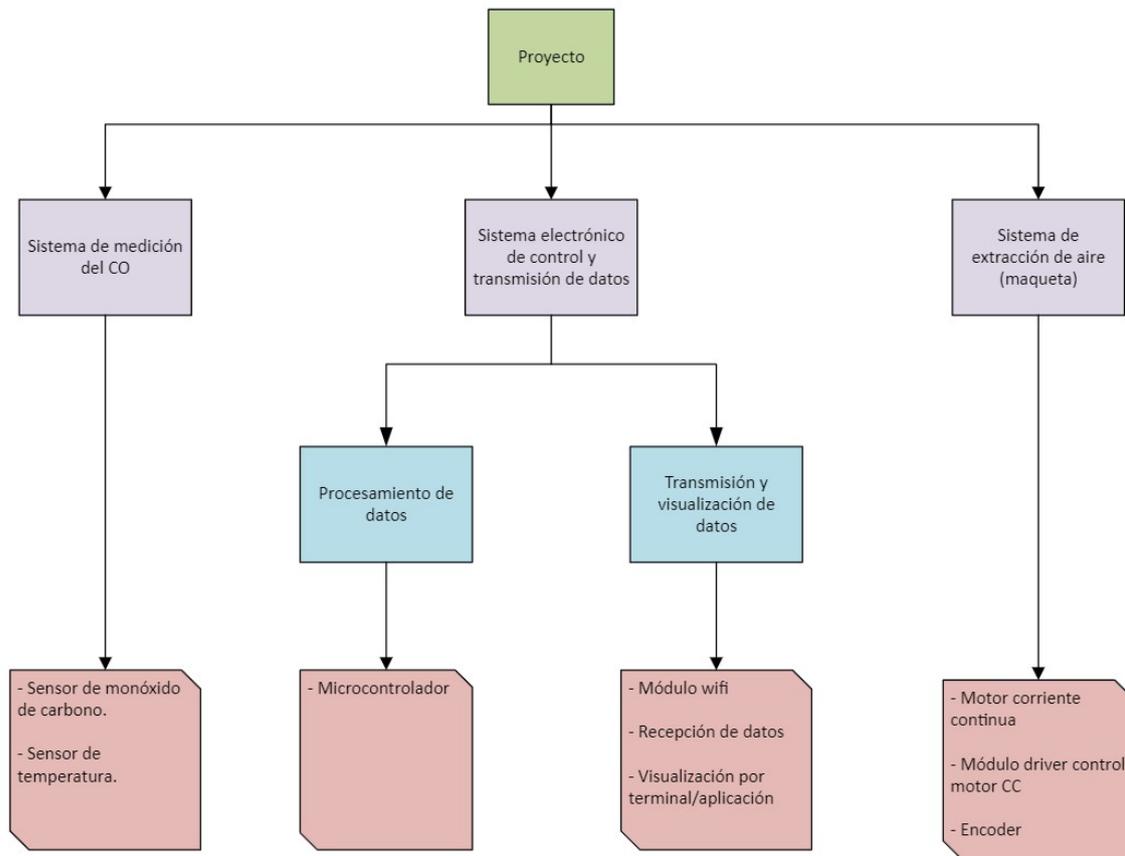


Figura 34. Estructura desglosada del proyecto (elaboración propia)

El proyecto completo se divide en 3 subsistemas, el de medición y toma de datos, el de control y transmisión y el de extracción de aire.

- **Subsistema 1. Sistema de medición del monóxido de carbono.** Este subsistema incluye aquellos componentes necesarios para la toma de medida del CO en el aire. Entre ellos encontramos el propio sensor de monóxido de carbono, contenido en un módulo para facilitar la conexión, y un sensor de temperatura para calibrar la medida.
- **Subsistema 2. Sistema de electrónico de control y transmisión de datos.** Su función es la de recibir la señal captada por el subsistema anterior y transformarla en un valor digital con el que se pueda trabajar. Está dividido en dos grupos, uno de procesamiento y otro de transmisión.
 - **Grupo 1: Procesamiento de datos.** Formado por un microcontrolador, es la parte encargada de la gestión de la información. Toma los datos obtenidos por el sensor, los transforma a una señal digital y, a partir de lo que se le haya programado, hace actuar al sistema de un modo u otro.

- **Grupo 2: Transmisión y visualización de datos.** Para mostrar las mediciones de gas al usuario final, el proyecto cuenta con un módulo wifi conectado a la placa para transmitir la señal a otro dispositivo, además de un método visual de lectura de información.
- **Subsistema 3. Sistema de extracción de aire (maqueta).** Al recibir una señal de activación por parte del microcontrolador, se debe encender el sistema de extracción de aire para la seguridad del usuario final. La extracción de aire se simula mediante un motor de corriente continua controlado por un módulo *driver* para motores de este tipo y cuenta con un *encoder* para medir su velocidad de giro.

5.1 Sistema de medición del monóxido de carbono.

Por sus características antes concretadas, especialmente su superior rango de detección con respecto al resto de opciones (de 0 a 5000 ppm), se ha optado por el uso del sensor de CO TGS5141.

Como se ha comentado, este sensor se puede encontrar en diversos módulos en el mercado. Estos módulos están diseñados para que su conexión con otros dispositivos, especialmente de control, sea sencilla, puesto que la electrónica interna está incluida. Como en el estudio de soluciones alternativas se ha determinado que la placa a utilizar era la STM32L053R8, se ha buscado un módulo que se adaptara a ella.

STMicroelectronics, los fabricantes de la gama de microcontroladores STM32 y de la *board* elegida, tienen en su catálogo unos módulos específicos, las placas de expansión P-NUCLEO-IKA02A1, con sensores electroquímicos pensados para la conexión con sus placas de desarrollo. Entre estos módulos hay un modelo disponible con el sensor de gas escogido.



Figura 35. Placa STM32L053R8 junto al módulo de expansión P-NUCLEO-1KA02A1 ²⁷

Esta placa de expansión está diseñada para sensores de diferentes tamaños, entre ellos el TGS5141, que viene incluido. Además de este, contiene dos amplificadores operacionales TSU111 de bajo consumo (precisan solo 900 nA para funcionar) y alta precisión (150 μ V) para el acondicionamiento de señal y un sensor de temperatura analógico STLM20 de 2.4 V que se usa para compensar las lecturas de gas [21].

Se ha seleccionado este módulo en concreto por dos razones principales: la incorporación del sensor de monóxido de carbono elegido y su adaptabilidad y sencillez para ser conectado con las placas de STM32.

A continuación, se detallan las especificaciones del sensor de monóxido de carbono elegido, el TGS5141, contenido en la placa.

²⁷ Extraída de: <https://www.st.com/en/ecosystems/p-nucleo-ika02a1.html>

Item	Specification
Model number	TGS5141-P00
Target gases	Carbon monoxide
Typical detection range	0 ~ 5,000ppm
Output current in CO	1.2~3.2nA/ppm
Baseline offset(*1)	<±10ppm equivalent
Operating temperature	-10°C ~ +50°C (continuous) -20°C ~ +60°C (intermittent)
Operating humidity	10 ~ 95%RH (no condensation)
Response time (T90)	within 60 seconds
Storage conditions	-10°C ~ +50°C (continuous) -20°C ~ +60°C (intermittent)
Weight	approx. 2.5g
Standard test conditions	20±2°C, 40±10%RH

Figura 36. Características básicas del sensor de gas TGS5141 [13]

Los detalles más relevantes son el amplio rango de detección ya mencionado, de 0 a 5000 ppm y el rango de salida de corriente del sensor: de 1,2 nA/ppm a 3,2 nA/ppm.

Por otro lado, también es interesante conocer los detalles del sensor analógico de temperatura STLM20 incorporado, ya que será necesario medir la temperatura para compensar su efecto sobre la medida de CO.

Según lo que indica su hoja de datos [22], es más preciso cuando está alimentado a 2,7 V y a temperatura ambiente y tiene un error lineal con un máximo en los extremos del rango de temperatura de $\pm 2,5$ °C.

Este sensor tiene que estar alimentado a un mínimo de 2,4 V y un máximo de 5,5 V en las condiciones normales de uso. Como está integrado en la placa, esto no será un problema [23]. Su tensión de salida típica, además, es de 1,8639 V.

Sym	Description	Test condition ⁽¹⁾	Min	Typ ⁽²⁾	Max	Unit
V _{CC}	Supply voltage	T _A = -30 °C to 130 °C	2.4		5.5	V
		T _A = -55 °C to 130 °C or -40 °C to 85 °C	2.7		5.5	V
V _O	Output voltage	T _A = 0 °C		1.8639		V
	Second order parabolic equation Temperature to voltage error ⁽³⁾ V _O = (-3.88 x 10 ⁻⁶ * T ²) + (-1.15 x 10 ⁻² * T) + 1.8639 V	T _A = 125 °C to 130 °C			±2.5	°C
		T _A = 80 °C to 85 °C		±1.00	±2.1	°C
		T _A = 25 °C to 30 °C		±0.5	±1.5	°C
		T _A = 0 °C		±0.75	±1.9	°C
		T _A = -40 °C			±2.3	°C
		T _A = -55 °C			±2.5	°C
I _Q	Quiescent current	2.4 V ≤ V _{CC} ≤ 5.5 V		4.8	8	μA
	Sensor gain (temperature sensitivity or average slope), V _O = -11.77 mV/°C * T + 1.860 V	-30 °C ≤ T _A ≤ 100 °C	-11.4	-11.77	-12.2	mV/°C
	Non-linearity	-20 °C ≤ T _A ≤ 80 °C		±0.4		%
ΔI _Q	Change of quiescent current	2.4 V ≤ V _{CC} ≤ 5.5 V		0.7		μA
T _C V _O	Temperature coefficient of quiescent current			-11		nA/°C
I _{SD}	Shutdown current	V _{CC} ≤ 0.8 V		0.02		μA
Z _O	Output impedance	0 μA ≤ I _L ≤ 16 μA ⁽⁴⁾⁽⁵⁾			160	Ω
RegL	Load regulation ⁽⁶⁾				-2.5	mV
RegI1	Line regulation	2.4 V ≤ V _{CC} ≤ 5.0 V			3.3	mV/V
RegI2		5.0 V ≤ V _{CC} ≤ 5.5 V			11	mV

Figura 37. Características eléctricas del sensor analógico de temperatura STLM20 [22]

El mencionado rango de medición depende del voltaje suministrado: de -55 °C a 130 °C si se alimenta con una fuente de entre 2,7 V a 5,5 V y de -30 °C a 130 °C para una tensión mínima de 2,4 V. Asimismo, puesto que su corriente de suministro máxima en reposo es de solo 8 μA, el autocalentamiento del sensor es prácticamente insignificante.

Symbol	Parameter	Value	Unit	
T _{STG}	Storage temperature	-65 to +150	°C	
T _S LD ⁽¹⁾	Lead solder temperature for 10 seconds	260	°C	
V _O	Output voltage	-0.3 to V _{CC} + 0.3	V	
V _{CC}	Supply voltage	-0.2 to 6.5	V	
I _O	Output current	10	mA	
T _{J(max)}	Maximum junction temperature	150	°C	
θ _{JA}	Thermal resistance	SOT323-5L	331.4	°C/W
		UDFN-4L	160.2	°C/W
T _A ⁽²⁾	Ambient operating temperature	Grade 7	-55 to 130	°C
		Grade 9	-40 to 85	°C

Figura 38. Valores nominales máximos del sensor analógico de temperatura STLM20 [22]

Con respecto a sus valores máximos obtenidos en pruebas de estrés, puede trabajar a temperaturas de entre -65 °C a 150 °C, las cuales no deberían alcanzarse en condiciones normales de funcionamiento.

Por último, es interesante conocer cómo varía la tensión de salida del sensor en función de la temperatura:

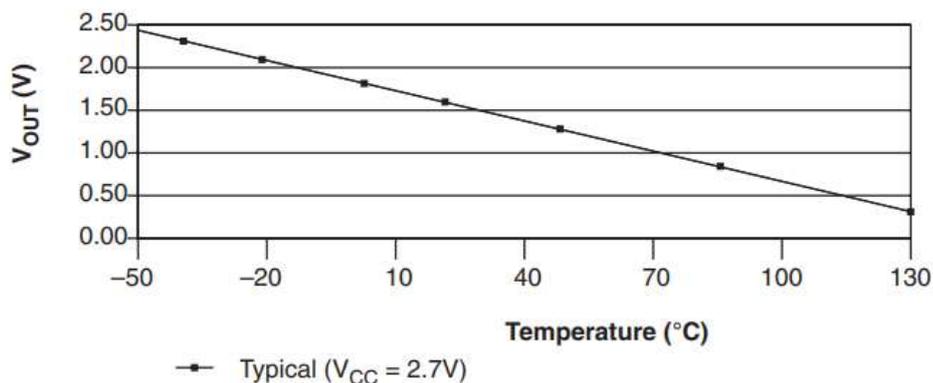


Figura 39. Tensión de salida del sensor STLM20 respecto a la temperatura [22]

5.2 Sistema electrónico de control y transmisión de datos.

En esta sección se describen los componentes y tecnologías que se usaran para tomar la señal analógica del sensor y enviarla por conexión wifi.

5.2.1 Microcontrolador

Como se ha mencionado en el apartado anterior, el microcontrolador elegido es el STM32L0, concretamente el incluido en la placa de desarrollo NUCLEO-L053R8. Es importante conocer las características y limitaciones de esta *board* para determinar las conexiones y selección del resto de componentes. Siguiendo los datos del *datasheet* del propio fabricante [23] encontramos los valores de funcionamiento:

Operating voltage	1.8 V to 3.6 V (down to 1.65 V at power-down) with BOR option 1.65 V to 3.6 V without BOR option
Operating temperatures	Ambient temperature: -40 to +125 °C Junction temperature: -40 to +130 °C

Figura 40. Valores de funcionamiento de la placa STM32L053R8 [23]

Vemos que la placa puede estar alimentada de 1.65 a 3.6 V. Cuanto menor sea la tensión de alimentación, menor será el consumo, pero tendrá limitadas algunas capacidades como muestra la figura siguiente:

Operating power supply range ⁽¹⁾	Functionalities depending on the operating power supply range		
	DAC and ADC operation	Dynamic voltage scaling range	USB
$V_{DD} = 1.65$ to 1.71 V	ADC only, conversion time up to 570 ksp/s	Range 2 or range 3	Not functional
$V_{DD} = 1.71$ to 1.8 V ⁽²⁾	ADC only, conversion time up to 1.14 Msps	Range 1, range 2 or range 3	Functional ⁽³⁾
$V_{DD} = 1.8$ to 2.0 V ⁽²⁾	Conversion time up to 1.14 Msps	Range 1, range 2 or range 3	Functional ⁽³⁾
$V_{DD} = 2.0$ to 2.4 V	Conversion time up to 1.14 Msps	Range 1, range 2 or range 3	Functional ⁽³⁾
$V_{DD} = 2.4$ to 3.6 V	Conversion time up to 1.14 Msps	Range 1, range 2 or range 3	Functional ⁽³⁾

Figura 41. Funcionalidades de la placa STM32L053R8 dependiendo de su rango de tensión de alimentación [23]

Por el momento, como la maqueta no está alimentada por baterías y el consumo no es problema, se alimentará a 3,3 V mediante uno de sus pines de suministro.

En cuanto al rango de temperatura, de -40 °C a 125 °C, el dispositivo no debería estar sometido a ambientes extremos, por lo que no se ha considerado una limitación.

Una vez consideradas las exigencias y restricciones de alimentación del microcontrolador, nos podemos centrar en las características de sus patillas.

El STM32L053R8 cuenta con 51 pines GPIO (*General Purpose Input Output*), es decir, 51 pines de entrada/salida de propósito general. El proyecto no va a utilizar tantos, así que queda hueco para la expansión con otros módulos o componentes si así se quisiera. Cada uno de los pines puede ser programado como salida (*push-pull* u *open-drain*) o entrada (con o sin *pull-up* o *pull-down*).

Peripheral		STM32L053C6	STM32L053R6	STM32L053C8	STM32L053R8
Flash (Kbytes)		32		64	
Data EEPROM (Kbytes)		2		2	
RAM (Kbytes)		8		8	
Timers	General-purpose	3		3	
	Basic	1		1	
	LPTIMER	1		1	
RTC/SYSTICK/IWDG/WWDG		1/1/1/1		1/1/1/1	
Communication interfaces	SPI/I2S	4(2) ⁽¹⁾ /1		4(2) ⁽¹⁾ /1	
	I ² C	2		2	
	USART	2		2	
	LPUART	1		1	
	USB/(VDD_USB)	1(1)		1(1)	
GPIOs		37	51 ⁽²⁾	37	51 ⁽²⁾

Figura 42. Funcionalidades, componentes y periféricos de la placa STM32L053R8 [23]

Respecto a la comunicación, tiene capacidad para distintas interfaces. En este caso, se va a extender sobre la USART (*Universal Synchronous and Asynchronous serial Receiver and Transmitter*), que es la que se va a usar para la comunicación con el módulo wifi. Esta placa cuenta con dos interfaces USART (USART1 y USART2) con velocidad de comunicación de hasta 4 Mbit/s.

5.2.1.1 UART/USART

USART es un protocolo que se utiliza para la comunicación entre distintos procesadores, igual que el UART, el I2C y el SPI, también incluidos en la placa. Este protocolo permite comunicar dos dispositivos distintos de manera simple, ya que solo utiliza dos líneas entre ellos para transmitir y recibir datos en ambas direcciones. Para comunicar los dispositivos, por tanto, basta con conectar el pin TX (transmisión) del primer dispositivo al pin RX (recepción) del segundo y viceversa, conectando también las masas entre ellos.

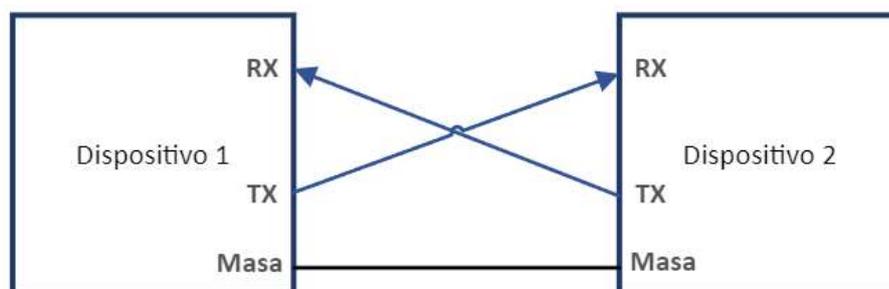


Figura 43. Esquema funcionamiento comunicación USART/UART (elaboración propia)

La comunicación USART es en serie y tanto síncrona como asíncrona, al contrario que la UART, que realiza solo comunicación en serie asíncrona.

Respecto a estos términos, la comunicación en serie consiste en enviar los datos bit a bit secuencialmente, utilizando solo una línea para comunicar y transferir datos. Es la forma opuesta de comunicación a la paralela, que utiliza múltiples líneas, ya que todos los bits se envían a la vez (un canal paralelo de 8 bits transmitirá los 8 al mismo tiempo, uno serie lo hará uno detrás de otro). Por tanto, para este protocolo de comunicación, los datos, que se almacenan en paralelo, se transmiten en serie desde el pin TX del transmisor hasta el pin RX del receptor y después se vuelven a convertir a paralelos.

A pesar de que con la comunicación paralela se pueden enviar más bits al mismo tiempo, y por tanto transmitir datos más rápidamente, puede tener problemas de sincronización al enviar muchos bits, sobre todo con cables largos, además de necesitar más líneas. Por esta razón, la comunicación serie es el modo más común de los dos, especialmente en proyectos de microcontroladores. Para compensar el hecho de que su rendimiento para transmitir datos es peor para una misma frecuencia, la comunicación serie se suele realizar a frecuencias más altas que la paralela.

Por otro lado, la comunicación serial tiene dos subclases: síncrona y asíncrona. Como se ha comentado anteriormente, la interfaz USART puede utilizar ambas.

En la comunicación asíncrona, el dispositivo transmisor y el receptor no tienen la misma señal de reloj así que tienen que transmitir a la misma velocidad para mantener la misma temporización de los bits.

Debido a que no se utiliza el reloj, el dispositivo transmisor precisa de una manera para señalar el inicio y fin del envío de datos. Para ello, se añade un bit de inicio (paso a nivel bajo) y otro de parada (nivel alto) para representar el inicio y el final de un mensaje, o trama de datos, e indicar al receptor cuándo comenzar y dejar de leer bits. Cuando el receptor detecta un bit de inicio, leerá los bits equivalentes a la tasa de baudios (número de bits por segundo) definida.

Por el contrario, en la comunicación síncrona, el dispositivo transmisor y el receptor están coordinados en el tiempo y los datos transmiten ininterrumpidamente sin añadir ningún bit adicional de sincronización. Para enviar los bits se utilizan pulsos de reloj complementarios que permiten tener sincronizados los dos extremos.

Para concluir, el protocolo USART (igual que el UART) es de uso común en proyectos de baja velocidad y poco rendimiento, puesto que es muy sencillo de utilizar e integrar, es estándar y económico. Esta es la razón principal de su selección.

5.2.1.2 Entorno de programación y librería HAL

Existen diversos entornos de desarrollo integrados (IDEs) y *debuggers* disponibles, así como otras herramientas de software, para configurar, programar o controlar los microcontroladores de la familia STM32, tanto de la propia compañía que los diseña como de terceros. Aunque la mayoría serían válidos para la realización del proyecto entre manos, en este caso se va a utilizar el IDE Keil μ Vision para la programación del micro.

Keil μ Vision 5 es una potente herramienta de software diseñada por Arm que tiene todas las características necesarias para el desarrollo de sistemas embebidos, como un editor de código, compilador de C/C++, *macro assembler*, *linker* y generador de ficheros HEX [24].

Cuenta con dos modos: modo construcción, para la creación de proyectos y aplicaciones y modo *debug*, para su depuración y prueba. Esto último puede hacerse mediante el simulador que viene integrado o por medio de hardware.



Figura 44. Logo KEIL (compañía)²⁸

El IDE y *debugger* μ Vision IDE es una herramienta adecuada para desarrollar software, ya que incluye diversas características que hacen que la programación sea más rápida y cómoda. Esto, sumado a la familiaridad previa con el programa, ha sido la razón principal de la elección de este entorno de desarrollo.

Este IDE cuenta con la posibilidad de trabajar y gestionar librerías de software cómodamente. Entre ellas, podemos encontrar la librería HAL, que se usará a lo largo del trabajo. La biblioteca HAL (*Hardware Abstraction Layer*) STM32Cube contiene funciones simples para gestionar los pines de entrada y salida GPIO y los periféricos que puedan conectarse. HAL contiene una considerable cantidad de APIs (*Application Programming Interfaces*, interfaz de programación de aplicaciones) de alto nivel y orientadas a funciones que se utilizan para diseñar e integrar software en aplicaciones. El uso de esta librería simplifica en gran medida el programar la comunicación y conexión entre el microcontrolador y los periféricos conectados.

²⁸ Extraída de: https://en.wikipedia.org/wiki/Keil_%28company%29

5.2.2 Módulo wifi

Como se ha establecido en el apartado anterior de selección de componentes, la placa elegida, STM32L053R8, cuenta únicamente con el microcontrolador STM32L0 el cual no tiene capacidad de conexión wifi. Para incorporarla, por tanto, habrá que hacer uso de un módulo externo, en concreto el ESP-01s, basado en el chip ESP8266EX.

El ESP8266 es un SoC (*System on Chip*) que salió al mercado en 2014 y cuya aparición supuso una revolución en el mundo del IoT, ya que era capaz de añadir funcionalidades wifi a proyectos que lo requirieran de forma mucho más barata que su competencia. Está diseñado y fabricado por la empresa china Espressif, igual que el ESP32 mencionado anteriormente, versión posterior de este dispositivo.



Figura 45. Chip ESP8266EX ²⁹

El chip utiliza el procesador Tensilica Xtensa LX106 y, entre algunas de sus principales características, destacan:

- Comunicación: cuenta con capacidades wifi 802.11 b/g/n de 2.4 GHz, I2C, SPI y pines dedicados a UART (Rx, Tx).
- Memoria: 64 KB de RAM de instrucciones y 96 KB de RAM de datos.
- Pines: cuenta con 16 pines GPIO.

Usualmente se suele encontrar este chip incorporado en módulos y placas de desarrollo, puesto que no cuenta con memoria Flash integrada, siendo el ESP-01 el primero en aparecer. Más tarde surgió la evolución de este módulo, el ESP-01s, muy similar en construcción y en términos de software, aunque con alguna diferencia en capacidad, conectividad y cableado interno.

²⁹ Extraída de: DigiKey ESP8266EX

Este módulo, además del chip ESP8266EX, cuenta con una memoria *flash* BG25Q80A donde se almacena el programa que se quiere ejecutar, un led azul que se activa si está encendido, una antena wifi y varios pines para la alimentación y comunicación. Estos son:

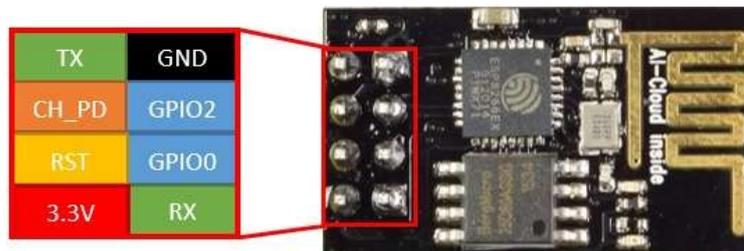


Figura 46. Conexiones módulo ESP01s³⁰

- GND.
- GPIO0 y GPIO2, pines de entrada y salida de propósito general.
- RX para la recepción de datos y TX para la transmisión. También se pueden utilizar como pines digitales GPIO.
- CH_PD, EN (Enable), para apagar el módulo si está a nivel bajo (0 V) y encenderlo si está a nivel alto (3,3 V).
- RST (RESET), reinicia el ESP-01s si está a nivel bajo.
- Vcc, para la alimentación. Funciona a 3,3 V y admite un máximo de 3,6 V. Un aspecto importante es que la corriente suministrada tiene que ser mayor de 200 mA para que algunas funciones wifi se ejecuten correctamente.

Por construcción, la memoria ROM (*Read Only Memory*, memoria de solo lectura) del ESP8266 viene con el firmware AT pre-instalado, por ello, puede ser controlado mediante comandos AT a través del puerto serie.

5.2.2.1 Comandos AT

El conjunto de comandos AT es como se denomina a un lenguaje creado en 1981 a partir del set de comandos de Hayes que se utiliza a menudo para operaciones de telecomunicaciones, como la parametrización y configuración de módems.

³⁰ Extraída de: <https://programarfacil.com/podcast/como-configurar-esp01-wifi-esp8266/>

Las instrucciones se envían por una interfaz estándar escogida, UART en este caso, para programar el módulo. La comunicación funciona en ambos sentidos, obteniendo respuesta para las instrucciones que se le envían (normalmente un “OK”).

“AT”, los caracteres que se utilizan como prefijo de los distintos comandos significa “Atención”, de ahí que se les conozca por ese nombre. Respecto a estos comandos, la mayoría (aunque no todos), contiene 4 tipos:

Tipo	Formato del comando	Descripción
Test	AT+<x>=?	Consulta el estado de parámetros internos o del comando set.
Consulta	AT+<x>?	Sirven para leer el valor actual del parámetro consultado.
Set	AT+<x>=<...>	Se utilizan para incluir algún parámetro y configurar el módulo.
Ejecutar	AT+<x>	Se realiza una función que ya viene incluida en el software del módulo.

Tabla 5: Tipos de comandos AT del chip ESP8266

Como se ha comentado, los chips ESP8266 vienen con el firmware AT ya incorporado de fábrica, así que se pueden utilizar comandos de diferentes clases para configurar y controlar el módulo wifi. Entre ellos, comandos básicos de funcionamiento, comandos wifi, comandos TCP/IP, HTTP, Web server, entre otros. A continuación, se describirán aquellos que se van a utilizar en este proyecto, el resto se puede encontrar en la documentación de Espressif, el fabricante [25].

Command	Description
AT	Test AT startup
AT+RST	Restart module
AT+GMR	View version info
AT+GSLP	Enter deep-sleep mode
ATE	AT.commands echo or not
AT+RESTORE	Factory Reset
AT+UART	UART configuration, [@deprecated]
AT+UART_CUR	UART current configuration
AT+UART_DEF	UART default configuration, save to flash
AT+SLEEP	Sleep mode
AT+RFPOWER	Set maximum value of RF TX Power
AT+RFVDD	Set RF TX Power according to VDD33

Figura 47. Listado de algunos de los comandos AT básicos para manejar el chip ESP8266 [25]

Command	Description
AT+CWMODE	Wi-Fi mode(sta/AP/sta+AP), [@deprecated]
AT+CWMODE_CUR	Wi-Fi mode(sta/AP/sta+AP) Won't save to Flash
AT+CWMODE_DEF	Wi-Fi default mode(sta/AP/sta+AP) Save to Flash
AT+CWJAP	Connect to AP, [@deprecated]
AT+CWJAP_CUR	Connect to AP, won't save to Flash
AT+CWJAP_DEF	Connect to AP, save to Flash
AT+CWLAPOPT	Set the configuration of command AT+CWLAP
AT+CWLAP	Lists available APs
AT+CWQAP	Disconnect from AP

Figura 48. Listado de algunos comandos AT para controlar las funcionalidades wifi del chip ESP8266 [25]

- AT+RST

AT+RST	
Función	Reinicia el módulo.
Tipo	Ejecutar
Respuesta	OK

Tabla 6: Resumen comando AT+RST del chip ESP8266

- AT+CWMODE_CUR =<mode>

AT+CWMODE_CUR=<mode>	
Función	Configura el modo de funcionamiento de entre uno de los 3 disponibles: softAP (soft Access Point), Station mode o ambos.
Tipo	Set
Respuesta	OK
Parámetros	Para <mode>: <ul style="list-style-type: none"> - 1: station mode - 2: softAP mode - 3: softAp + station mode

Tabla 7: Resumen comando AT+CWMODE_CUR del chip ESP8266

Respecto a los distintos modos que se configuran con este comando, en el modo Station (estación), el ESP8266 se conectará a una red wifi ya configurada por un punto de acceso, como un rúter. Para hacerlo, se ha de indicar posteriormente mediante otro comando a qué red conectarse.

El modo Access Point (punto de acceso), por el contrario, configura el módulo para actuar como un punto de acceso que proporciona una red wifi a la que pueden conectarse otros dispositivos. Como normalmente los puntos de acceso proporcionan conexión a internet mediante el uso de cables y este no es el caso, se denomina Soft Access Point.

Respecto al tercer modo, el ESP8266 funciona tanto como estación como punto de acceso.

En la versión más reciente del firmware, el comando AT+CWMODE=<mode> es sustituido por dos con funcionamiento similar: AT+CWMODE_CUR=<mode>, que no guarda la configuración seleccionada en la memoria flash y AT+CWMODE_DEF=<mode>, que sí lo hace. CUR hace referencia a *current* (actual) y DEF a *default* (por defecto).

Los comandos AT que empiezan con el prefijo "CW" en el ESP8266 sirven para configurar y manejar su conectividad wifi: "C" suele representar la conexión o configuración y "W" el wifi. De esta variedad de comandos "CW" se va a utilizar alguno más, como se puede leer a continuación.

- AT+CWJAP =<mode>

AT+CWJAP=<ssid>, <pwd>	
Función	Conectarse a un punto de acceso
Tipo	Set
Respuesta	OK
Parámetros	<ssid>: string, el SSID del punto de acceso <pwd>: string, la contraseña del punto de acceso

Tabla 8: Resumen comando AT+CWJAP del chip ESP8266

Si el módulo se encuentra en modo estación puede conectarse a un punto de acceso dentro de su rango. Para ello hay que indicarle su SSID (*Service Set Identifier*, identificador de red) y contraseña. Si alguna de las opciones no es correcta devolverá un mensaje de error diferente.

Para que funcione, los comandos AT que utilizan *strings* precisan que estén escritos con comillas dobles, por ejemplo: AT+CWJAP="nombre_wifi", "contraseña_wifi".

Este comando, por tanto, servirá para conectar el módulo wifi al rúter del lugar donde se quiere utilizar.

- AT+CWQAP =<mode>

AT+CWQAP	
Función	Desconectarse de un punto de acceso
Tipo	Ejecutar
Respuesta	OK

Tabla 9: Resumen comando AT+CWQAP del chip ESP8266

Este comando es el opuesto al de AT+CWJAP, ya que sirve para desconectar al chip del punto de acceso al que esté conectado. Si funciona correctamente devuelve el mensaje "OK" y si el ESP actualmente no está conectado a ningún AP devuelve "QAP fail" y "ERROR". Se utilizará en este caso para asegurarse de que el módulo no se encuentra conectado a ninguna estación previo a su conexión.

- AT+CIFSR

AT+CIFSR	
Función	Devuelve la dirección IP local del módulo como cliente.
Tipo	Ejecutar
Respuesta	(direcciones IP) + OK

Tabla 10: Resumen comando AT+CIFSR del chip ESP8266

Con este comando se puede obtener la dirección IP asignada al módulo si está conectado a una red wifi. La respuesta que se obtiene depende del modo en que esté funcionando el módulo (seleccionado con el comando AT+CWMODE o sus variantes).

- Dirección IP Estación (STAIP): Esta es la dirección IP asignada al ESP8266 cuando está conectado a una red wifi como cliente (modo estación), es decir, la dirección que utiliza para comunicarse con otros dispositivos o servidores en la red.
- Dirección IP Punto de Acceso (AP): Esta es la dirección IP del punto de acceso wifi que ha creado el ESP8266 si está en modo *Acces Point* y otros dispositivos se pueden conectar a él.

```
AT+CIFSR
+CIFSR:STAIP,"192.168.1.74"
+CIFSR:STAMAC,"08:3a:8d:de:ed:87"
OK
```

Figura 49. Ejemplo de respuesta al comando AT+CIFSR de un chip ESP266 conectado en modo punto de acceso

- AT+CIPSTART

AT+CIPSTART	
Función	Establecer una conexión TCP, UDP o SSL
Tipo	Set
Respuesta	OK, ERROR o ALREADY CONNECT
Parámetros	<enlace ID>: ID de la conexión de red (0-4), en caso de que haya más de una. <tipo>: string, "TCP", "UDP" o "SSL". <IP remota>: string, dirección IP remota. <puerto remoto>: string, número de puerto remoto. (Parámetros adicionales propios de cada tipo de conexión).

Tabla 11: Resumen comando AT+CIPSTART del chip ESP8266

Este comando permite establecer una conexión de tipo TCP, UDP o SSL, dependiendo del parámetro que se le pase.

Escribiendo la IP y el puerto de un servidor abierto con el mismo protocolo de comunicación, el módulo se conectará como cliente.

- AT+CIPMODE

AT+CIPMODE	
Función	Establecer el modo de transferencia de datos.
Tipo	Set
Respuesta	OK
Parámetros	0: modo normal 1: modo passthrough (paso) para UART-wifi.

Tabla 12: Resumen comando AT+CIPMODE del chip ESP8266

El comando AT+CIPMODE permite seleccionar entre dos modos: el 0, o normal, y el 1, o modo de paso, para el protocolo UART-wifi. En este segundo caso se habilita la denominada transmisión transparente, en la cual se transmiten los datos desde la fuente hasta el destino sin ningún cambio en la comunicación [26].

Si el ESP8266 está conectado a un servidor TCP (comando AT+CIPSTART) y le envía información, el modo de paso hace que los mensajes lleguen sin necesidad de utilizar el comando de envío cada vez. Para dejar de enviar datos y volver al envío de comandos AT hará falta mandar un paquete que incluya únicamente los caracteres “+++”.

- AT+CIPSEND

AT+CIPSEND	
Función	Enviar datos (a servidor).
Tipo	Ejecutar
Respuesta	Tras ejecutar el comando, aparece el carácter ">" para comenzar a recibir los datos. ERROR si no puede establecer conexión. SEND OK si los datos se han transmitido correctamente.
Parámetros	<enlace ID>: ID de la conexión (0-4), en caso de que haya más de una. <tamaño>: tamaño del dato a enviar (máx. 2048 bytes).

Tabla 13: Resumen comando AT+CIPSEND del chip ESP8266

Este es el comando básico para la transmisión de información. Tiene dos parámetros principales: el identificador de la conexión a la que se quiere enviar la información, opcional si no hay más de una, y el tamaño del dato a enviar, de máximo 2048 bytes.

Al ejecutarlo, aparece el carácter ">" por pantalla y, a continuación, es posible escribir el mensaje, que se enviará una vez alcanzada la longitud determinada.

Si se ha establecido una transmisión de datos transparente, como la que se ha definido anteriormente en el comando AT+CIPMODE, no es necesario establecer longitud. El chip comienza a enviar los paquetes datos que recibe, con un Intervalo de 20ms por paquete, sin necesidad de escribir el comando SEND cada vez [26].

5.3 Visualización de la información e interfaz gráfica

Es vital que el usuario final del dispositivo tenga conocimiento de los niveles de monóxido de carbono en la estancia donde haya decidido colocar el detector. Para ello, puesto que este aparato no cuenta con ningún tipo de *display* o método de visualización integrado, el microcontrolador ha de tener la capacidad de enviar los datos inalámbricamente a un terminal que pueda recibirlos y mostrarlos por pantalla.

Se va a diseñar, por tanto, un programa simple con el IDE QT Creator que se encargue de abrir desde un ordenador un servidor en configuración TCP/IP al cual el chip wifi se deberá poder conectar a modo de cliente.

5.3.1 TCP/IP

TCP/IP es un conjunto de protocolos que se utiliza para comunicar las diferentes máquinas que pertenecen a una red.

Las siglas TCP/IP significan Protocolo de Control de Transmisión/Protocolo de Internet. En este caso, el protocolo se refiere al conjunto de normas que definen qué formato han de tener los mensajes y los procedimientos para que puedan ser interpretados correctamente al enviar y transmitir información entre máquinas y programas [27].

Para garantizar que las comunicaciones lleguen sin imprecisiones, el modelo TCP/IP divide los datos en paquetes y los trata siguiendo un método de cuatro capas. Los datos las atraviesan al ser enviados y luego las vuelven a atravesar en sentido contrario para recibirlos en el destino [27] [28].

- Nivel 1 (nivel físico o capa de acceso a red): gestiona la infraestructura física y el medio de transmisión, como cables ethernet, redes inalámbricas, tarjetas de red y *drivers*.

Esta capa también incluye la codificación de la señal que hace posible que dos dispositivos se "entiendan": ethernet (802.3), wifi (802.11), entre otros.

- Nivel 2 (capa de red, internet): controla el movimiento de los paquetes de datos alrededor de la red.

El objetivo es poder interconectar varias LANs (redes de área local) entre sí y encaminar los datos entre ellas para enviar información.

Para asegurar que los datos lleguen al destino correcto se identifican los dispositivos con direcciones del nivel de red (En IP se llaman direcciones IP).

- Nivel 3 (capa de transporte): genera una conexión entre dos dispositivos. Los protocolos de transporte TCP/IP identifican las aplicaciones que envían o reciben bytes dentro de un dispositivo mediante el uso de los puertos (identificador de 16 bits).

Existen dos protocolos a nivel de transporte sobre IP: UDP (*User Datagram Protocol*) basado en la transmisión de bloques de bytes y TCP (*Transmission Control Protocol*) basado en la transmisión de un *stream* de bytes.

- Nivel 4 (capa de aplicaciones): La capa de aplicaciones es el grupo de aplicaciones con las que el usuario final puede interactuar para acceder a la red. En esta capa se regula la comunicación de los programas, que sobre todo usan los protocolos HTTP y FTP.

El protocolo TCP es uno de los protocolos fundamentales de Internet, ya que permite que las aplicaciones puedan comunicarse de forma fiable. Esto se debe a varios factores, entre ellos, el hecho de que entrega los datos en orden y sin duplicados a la aplicación destino en forma de un *stream* de bytes (e informa del error a la aplicación cuando no es posible). Además, controla que la velocidad a la que la aplicación origen envía bytes no sea superior a la que la aplicación destino pueda leerlos

Otro aspecto favorable es que en caso de pérdidas o duplicados de segmentos, TCP evita interrumpir la comunicación si puede resolverlo.

Por las razones mencionadas, el protocolo TCP ha sido la opción principal para la comunicación wifi del proyecto, por encima del protocolo UDP, que, aunque puede ser más rápido, no garantiza la entrega de datos.

A la hora de configurarlo es preciso tener en cuenta que se deben conectar las dos aplicaciones que se vayan a comunicar antes de enviar datos. Asimismo, la comunicación es de tipo bidireccional, es decir, el TCP maneja dos *streams* de bytes, uno de cliente a servidor y otro de servidor a cliente para garantizar que se transmitan todos los datos.

En la arquitectura cliente-servidor para aplicaciones en red hay, al menos, un dispositivo, el servidor, que está siempre en funcionamiento, esperando que el resto, los clientes, se conecten a él para así ofrecerles un servicio. En el caso de este proyecto, se va a diseñar un servidor en QT Creator al cual será posible conectarse con el módulo wifi (cliente).

5.3.2 QT Creator

Qt Creator es un entorno de desarrollo integrado especialmente diseñado para desarrollar Interfaces Gráficas de Usuario (GUI), programas y aplicaciones para ordenadores, sistemas embebidos y teléfonos móviles, así como consolas para servidores que no precisan de interfaz. Incluye un editor de código y una plataforma para construir las interfaces gráficas a partir de los denominados *QT widgets* [29].

Qt utiliza el lenguaje de programación C++ y está especialmente planteado para la programación orientada a objetos QT. Esta se fundamenta en crear clases nuevas a partir de objetos de la clase base *QObject*, heredando así sus propiedades. Además, el entorno de desarrollo incluye diversas librerías que añaden muchas funcionalidades a C++ y facilitan la tarea de programar.



Figura 50. Logo framework QT³¹

Entre las clases desarrolladas por QT se van a destacar *QTCPServer*, *QTcpSocket*, *QHostAddress*, *QSqlDatabase* y *QSqlQuery*, que son las que se utilizarán en el programa diseñado para abrir la comunicación en un puerto, guardar los datos enviados por el módulo wifi en una base de datos y mostrarlos por pantalla al usuario.

- *QTCPServer*: sirve para crear un servidor TCP y abrir un puerto de comunicación al que se podrán conectar los clientes.
- *QTcpSocket*: para que alguna aplicación pueda utilizar protocolos TCP/IP, los distintos sistemas operativos ponen a su disposición una API (Interfaz de Programación de Aplicaciones), los sockets.

³¹ Extraída de: [https://es.wikipedia.org/wiki/Qt_\(biblioteca\)](https://es.wikipedia.org/wiki/Qt_(biblioteca))

En QT se hace con la clase QTcpSocket. Esta permite comunicar aplicaciones mediante el envío de *streams* de bytes. El servidor crea un objeto de esta clase por cada cliente que se vaya a conectar. [30]

- QHostAddress: sirve para definir direcciones IP (IPv4 e IPv6).
- QSqlDatabase: se utiliza para conectar, configurar y establecer una sesión con un sistema de gestión de bases de datos.
- QSqlQuery: Se utiliza para ejecutar comandos sobre un sistema de gestión de bases de datos. Permite el uso de métodos para acceder a los contenidos de una tabla con las consultas SQL (*Structured Query Language*, Lenguaje de Consulta Estructurado).

5.3.3 SQLite

Para guardar un registro de las mediciones que se van realizando, es útil contar con una base de datos. Desde QT es posible crear bases de datos diferentes, usando drivers específicos para comunicarse con ellas, tal y como se puede ver en la Tabla 14, extraída de su página web [31]:

Bases de datos compatibles con QT	
Driver	Sistema manejador de bases de datos (SGBD)
QDB2	IBM DB2
QIBASE	Borland InterBase / Firebird
QMYSQL / MARIADB	MySQL o MariaDB
QOCI	Oracle Call Interface Driver
QODBC	Open Database Connectivity (ODBC) - Microsoft SQL Server
QPSQL	PostgreSQL
QSQLITE	SQLite version 3
QMIMER	Mimer SQL

Tabla 14. Listado de bases de datos compatibles con QT y el nombre del driver para manejarlas

Algunos de estos drivers vienen incluidos al descargar el programa y otros se pueden añadir manualmente. SQLite, además de ser el sistema de gestión de bases de datos más probado y contar con soporte en todas las plataformas según la compañía, pertenece al primer grupo. Estos beneficios, sumados a la familiaridad previa y a su uso común, han sido la razón de su selección.

El sistema SQLite es de dominio público y está disponible en una biblioteca escrita en el lenguaje C. Esta se integra directamente en el programa desarrollado y se controla mediante llamadas a subrutinas y funciones, por lo que hay poco *delay* para acceder a los datos

almacenados. A diferencia de otros sistemas que utilizan la estructura de cliente-servidor, no necesita ningún software de servidor adicional para funcionar.

Como la base de datos SQLite se guarda como un único archivo en un dispositivo, se puede encontrar normalmente en aplicaciones sencillas que utilizan pocos datos, sistemas embebidos y dispositivos integrados IoT que requieren una base de datos incorporada, como es el caso del proyecto entre manos.

5.4 Extracción de aire

5.4.1 Controlador y motor

De entre las opciones que dispone ST en su catálogo para el manejo de motores desde placas de desarrollo de la marca, finalmente se ha optado por el módulo X-NUCLEO-IHM13A1.

La placa está basada en el controlador STSPIN250, de bajo voltaje, para un único motor DC con escobillas. Este incluye un puente H construido a partir de transistores MOSFET para cambiar el sentido de giro, control por corriente PWM, un consumo muy reducido y protecciones contra sobrecorrientes y sobrecalentamientos [32].

Con respecto al módulo, tiene una tensión de alimentación de 1,8 V a 10 V, un valor eficaz de intensidad de salida de hasta 2,6 A RMS y es compatible con las placas de desarrollo STM32, así como aquellos componentes que utilicen los conectores Arduino UNO R3. Tiene además conectores para la fuente de alimentación y el motor y varios leds integrados [33].

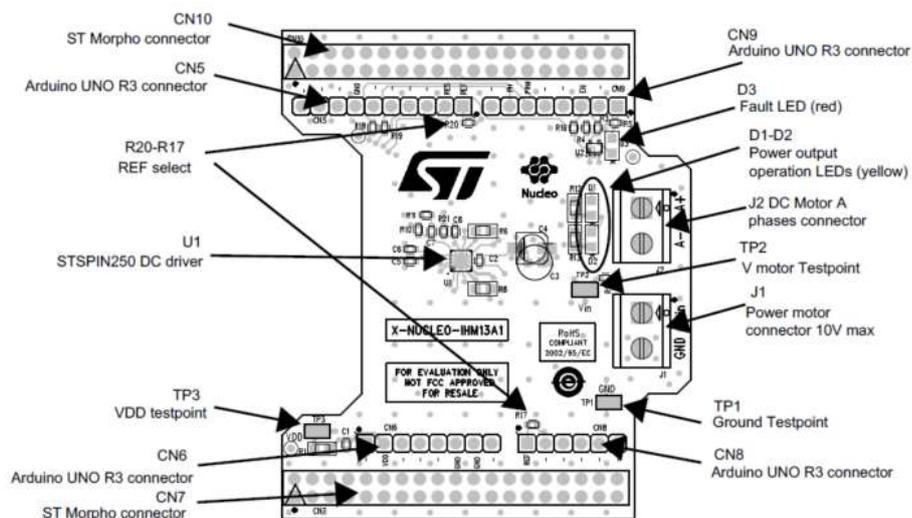


Figura 51. Esquema conexiones módulo X-NUCLEO-IHM13A1 [33]

En el punto de soluciones alternativas anterior se ha determinado que el motor se controlará variando el ancho de pulso de una señal PWM. Para generar este tipo de señales en placas STM32 se utilizan sus *timers* incorporados.

Estos temporizadores, que funcionan a modo de contador, se pueden sincronizar con el reloj interno, de modo que comienzan a contar hasta alcanzar su valor de registro de auto recarga. Cuando esto ocurre, el *timer* salta y se reinicia o activa una interrupción, en este caso poniendo el pin del canal de salida en nivel alto. Luego, cuando el temporizador alcanza otro valor determinado, el pin del canal de salida pasa a nivel bajo, volviendo a comenzar el ciclo. De esta manera, se acaba generando una onda PWM, cuya frecuencia está determinada por el reloj interno, el *prescaler*, y el registro de auto recarga.

5.4.2 Codificador rotatorio

Se ha determinado que para mensurar la velocidad a la que gira el motor se va a acoplar el *encoder* rotatorio AMT103-V. Este está diseñado para acoplarse en el extremo de un eje rotatorio, ya que cuenta con un engranaje que gira junto a él. Además, incluye 9 manguitos de colores para poder ajustarlo a ejes de los 2 a las 8 mm de diámetro.

Los codificadores de la serie AMT10 son de tipo incremental, es decir, generan una serie de pulsos eléctricos según se mueve el eje que se pueden utilizar para medir su velocidad, posición y dirección. Los pulsos se generan según la luz que miden unos fotodetectores, y esta varía dependiendo de cómo gire un disco que está delante de su fuente y conectado al eje que se quiere medir.

Los *encoders* incrementales disponen de 2 salidas (A y B) por las que emiten dos ondas cuadradas, desfasadas 90 grados cuando hay movimiento, lo que permite determinar el sentido de giro (horario o antihorario). Además, algunos tienen también una tercera salida índice de referencia que se puede utilizar para determinar un punto específico en la rotación o si el *encoder* ha dado una vuelta completa.

En este caso, el AMT103 dispone de 5 pines: canal A, B, índice (X), alimentación y masa. Además, contiene un *DIP switch* (interruptor DIP) con el que se pueden ajustar 16 valores de resolución, generándose 2048 pulsos de salida por revolución, o PPR, en la configuración por defecto (los cuatro interruptores en *off*).

Por otro lado, su tensión de entrada típica es de 5 V con un consumo de 6 mA, por lo que habrá que tener en cuenta estos requerimientos al conectarlo.

Puesto que su velocidad máxima de giro es de 15000 rpm, no habrá problema para conectarlo al motor, puesto que, sin carga, rota como máximo a 5000 rpm. Además, sus agujeros para el montaje permiten hasta 8 configuraciones distintas, de modo que tiene la posibilidad de utilizarse con gran variedad de motores.

6 DESARROLLO DEL PROYECTO. IMPLEMENTACIÓN

Habiéndose descrito los diferentes elementos del proyecto, se va a especificar cómo conectarlos y configurarlos para que funcionen como se precisa de ellos.

6.1 Cálculos y montaje del hardware

En esta sección se detalla cómo componer las partes físicas del sistema: sensor de monóxido de carbono, microcontrolador, módulo wifi y maqueta de extracción de aire.

6.1.1 Conexión y configuración del sensor de CO

En primer lugar, se deben ajustar los jumpers de la placa STM32. De acuerdo con su configuración estándar, deben estar en la siguiente disposición:

- JP1 desconectado
- JP5 (PWR) en el lado marcado con UV5
- JP6 (IDD) conectado

La conexión entre la placa de expansión y la de desarrollo es muy simple gracias al uso de los conectores de tipo Arduino UNO R3. La disposición de los pines entre placas coincide, así que bastaría con colocar la primera encima de la segunda. Puesto que la placa adquirida viene diseñada para el sensor elegido, no es necesario hacer ningún cambio en el sistema de acondicionamiento o en los puentes de soldadura presentes.

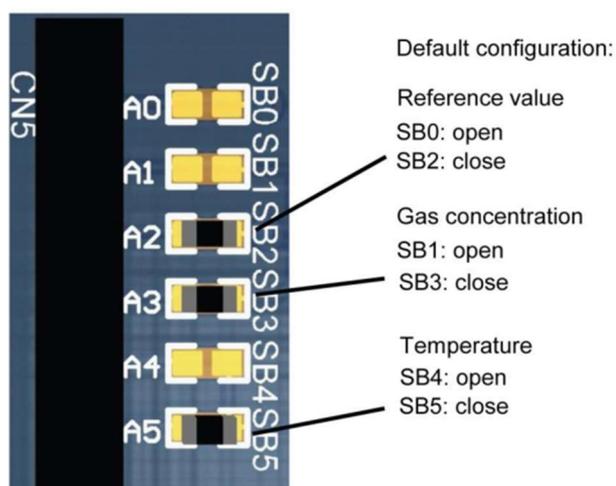


Figura 52. Configuración puentes de soldadura P-NUCLEO-IKA02A1 [21]

Por otro lado, la placa cuenta con dos jumpers accesibles, JP1 y JP2.

JP1 está presente debido a que la placa permite el uso de sensores con 2, 3 o 4 patillas así que hay que ajustar las conexiones. Para ello, se debe seguir las indicaciones de la ficha técnica del P-NUCLEO-IKA02A1 [34] con respecto al *jumper* JP1:

Sensor type	JP1 jumper
2 wires	Shorted
3 or 4 wires	Open
TGS5141	Doesn't matter

Figura 53. Colocación del jumper JP1 del módulo P-NUCLEO-IKA02A1 para los distintos sensores compatibles [34]

Para el sensor TGS5141, incorporado en este caso, no tiene importancia el estado de esta conexión, así que se ha dejado abierto.

Por su parte, la señal de salida de corriente del sensor es muy pequeña, así que debe ser amplificada por un amplificador operacional para conseguir suficiente resolución al convertir a voltaje. Debido a que cada sensor electroquímico que puede tener incorporada la placa puede producir una señal de corriente diferente, JP2 se utiliza para ajustar la ganancia en la fase de acondicionamiento. La hoja de datos [34] muestra la configuración por defecto:

Position	Gain	Capacitor value	Max. sensor current
Pins 1 and 2 shorted	47 k	100 n	60 μ A
Pins 2 and 3 shorted	470 k	1 μ	6 μ A

Figura 54. Colocación del jumper JP2 del módulo P-NUCLEO-IKA02A1 para ajustar la ganancia [34]

Para que funcione con el condensador de un microfaradio recomendado por el fabricante [13], hay que cortocircuitar los pines 2 y 3 con el *jumper* JP2 y así trabajar con una ganancia de 470000, es decir, que la red resistiva del amplificador cuente con una resistencia de 470000 k Ω y un condensador de 1 μ F.

Dicho amplificador operacional funciona en modo de amplificador de transresistencia, o convertidor de corriente a tensión, ya que se encarga de convertir la corriente de entrada proveniente del sensor de CO en una tensión proporcional a la salida. Se utiliza esta configuración debido a que el sensor entrega poca corriente, así que se puede transformar a una señal de tensión proporcional mayor que puede ser leída mediante uno de los pines analógicos de la placa.

Además, para evitar que el amplificador operacional TSU111 se sature a nivel mínimo cuando no haya presencia de gas, se va a utilizar un voltaje de referencia V_{REF} de 430 mV.

Por tanto, el circuito de amplificación de la señal y conversión a entrada de voltaje queda así:

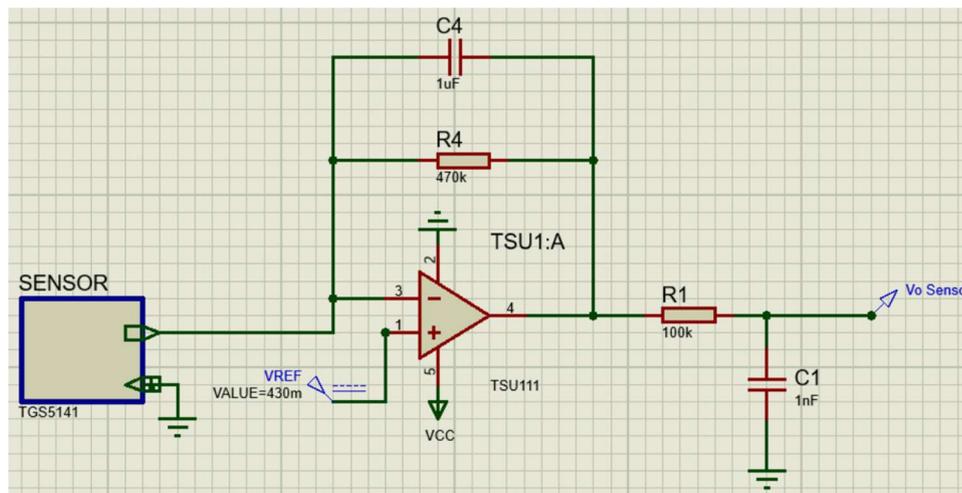


Figura 55. Circuito de amplificación y conversión a voltaje de la señal proveniente del sensor TGS5141 (elaboración propia)

Por otro lado, consultando las notas de aplicación del fabricante para la serie de sensores TGS5xxx [35], se obtiene la función necesaria para el cálculo de concentración de gas CO:

$$C = \frac{(V_{out} - V_o)}{\frac{A}{\alpha t}} = \frac{(V_{out} - V_o) \cdot \alpha t}{A}$$

Con C la concentración de CO en ppm, V_{out} la salida del sensor en voltios, V_o el voltaje de offset, A la ganancia y αt la sensibilidad al monóxido de carbono compensada con la temperatura.

Sin embargo, dependiendo del amplificador operacional que se use, el voltaje de offset puede tener una mayor dependencia de la temperatura. Por ello, se puede ajustar la expresión del cálculo de concentración para considerar este efecto:

$$C = \frac{(V_{out} - V_o(T))}{\frac{A}{\alpha t}} = \frac{(V_{out} - V_o(T)) \cdot \alpha t}{A}$$

La mencionada sensibilidad compensada se puede calcular con la siguiente ecuación:

$$\alpha t = \alpha / CF$$

En ella, CF es un coeficiente que se utiliza para compensar la influencia de la temperatura en la medida y que también se puede encontrar en la ficha técnica [35].

Temp (°C)	CF (l/l _o)	Temp (°C)	CF (l/l _o)	Temp (°C)	CF (l/l _o)
	TGS5141-NP0/PP0		TGS5141-NP0/PP0		TGS5141-NP0/PP0
-40	0.1589	0	0.7835	40	1.1500
-39	0.1761	1	0.7944	41	1.1575
-38	0.1933	2	0.8053	42	1.1650
-37	0.2105	3	0.8162	43	1.1725
-36	0.2277	4	0.8271	44	1.1800
-35	0.2448	5	0.8380	45	1.1875
-34	0.2620	6	0.8489	46	1.1950
-33	0.2792	7	0.8598	47	1.2025
-32	0.2964	8	0.8707	48	1.2100
-31	0.3136	9	0.8816	49	1.2175
-30	0.3308	10	0.8925	50	1.2250
-29	0.3479	11	0.9034	51	1.2271
-28	0.3651	12	0.9143	52	1.2292
-27	0.3823	13	0.9252	53	1.2313
-26	0.3995	14	0.9361	54	1.2334
-25	0.4167	15	0.9470	55	1.2355
-24	0.4339	16	0.9579	56	1.2376
-23	0.4510	17	0.9688	57	1.2396
-22	0.4682	18	0.9797	58	1.2417
-21	0.4854	19	0.9906	59	1.2438
-20	0.5026	20	1.0000	60	1.2459
-19	0.5198	21	1.0075	61	1.2516
-18	0.5370	22	1.0150	62	1.2574
-17	0.5542	23	1.0225	63	1.2632
-16	0.5713	24	1.0300	64	1.2690
-15	0.5885	25	1.0375	65	1.2748
-14	0.6057	26	1.0450	66	1.2806
-13	0.6229	27	1.0525	67	1.2864
-12	0.6401	28	1.0600	68	1.2922
-11	0.6573	29	1.0675	69	1.2980
-10	0.6745	30	1.0750	70	1.3038
-9	0.6854	31	1.0825		
-8	0.6963	32	1.0900		
-7	0.7072	33	1.0975		
-6	0.7181	34	1.1050		
-5	0.7290	35	1.1125		
-4	0.7399	36	1.1200		
-3	0.7508	37	1.1275		
-2	0.7617	38	1.1350		
-1	0.7726	39	1.1425		

Figura 56. Coeficientes de compensación del CO con la temperatura del sensor TGS5141 [35]

Aplicando la expresión, para el cálculo de concentración de monóxido de carbono ambiental compensado con la temperatura se utilizará la ecuación siguiente:

$$C = \frac{(V_{out} - V_o(T)) \cdot \alpha}{A \cdot CF}$$

La medida de concentración de monóxido de carbono tiene cierta dependencia de la temperatura a la que está sometido el sensor utilizado, tal y como se puede ver en la expresión anterior y en la siguiente gráfica (Figura 57) extraída de su hoja de datos [35].

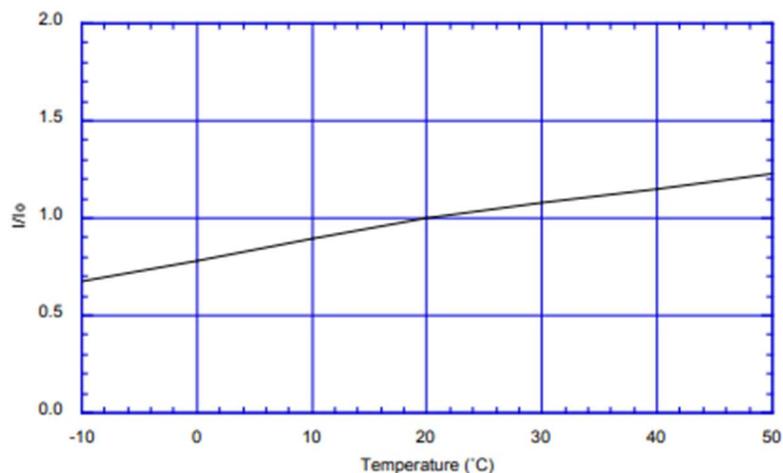


Figura 57. Dependencia de la temperatura en la medida de CO del sensor TGS5141

En ella, el eje de abscisas corresponde a una temperatura de $-10\text{ }^{\circ}\text{C}$ a $50\text{ }^{\circ}\text{C}$ y el eje de ordenadas al ratio $\frac{I}{I_o}$, siendo I la salida en corriente del sensor para 400 ppm de CO a varias temperaturas e I_o la salida para la misma concentración a $20\text{ }^{\circ}\text{C}$ y 50% de humedad relativa. Visualmente se comprueba el efecto de la temperatura sobre la señal, así como la relación lineal entre el ratio de salida y la concentración, constante independientemente del rango de concentración de CO.

Por tanto, es necesario saber el valor de la temperatura ambiente para conseguir una medida precisa del CO. De acuerdo con su *datasheet* [22], la función de transferencia del sensor de temperatura STLM20, incluido en el módulo, puede expresarse con diferentes niveles de precisión: bien con una ecuación de primer orden lineal o bien con una de segundo orden parabólica.

- Ecuación de primer orden:

$$V_o = -11,69 \cdot 10^{-3} \cdot T + 1,18663 \text{ (V)}$$

- Ecuación de segundo orden:

$$V_o = (-3,88 \cdot 10^{-6} \cdot T^2) + (-1,15 \cdot 10^{-2} \cdot T) + 1,8639 \text{ (V)}$$

Puesto que la ecuación parabólica es más precisa, es la que se utilizará cuando se mida la temperatura ambiente. Despejando su valor en esta segunda ecuación se obtiene:

$$T = -1481,96 + \sqrt{2,1926 \cdot 10^6 + \frac{(1,8639 - V_o)}{3,88 \cdot 10^{-6}}} \text{ (}^{\circ}\text{C)}$$

Puesto que la placa viene preconfigurada para utilizar los pines de entrada analógicos A2, A3 y A5, estando soldados y conectados con resistencias, se van a utilizar estos mismos para medir el voltaje de referencia, la temperatura y el monóxido de carbono, por lo que se deberán configurar como entradas analógicas más adelante (6.2.1.1 STM32CubeMX).

6.1.1.1 Prueba de funcionamiento del sensor de monóxido de carbono

Una vez conectada la placa, es recomendable comprobar que el sensor funciona. Para ello, se puede elaborar un código simple con las funciones de transferencia anteriores y utilizar alguna de las librerías especializadas que ST tiene disponibles en su web para descargar de forma libre y gratuita, el X-CUBE-IKA02A1 paquete de expansión de software. Entre ellas, se aprovecharán las que contienen funciones para la calibración del sensor y la lectura de entradas analógicas.

Anteriormente se ha mencionado que la concentración de gas CO depende de la sensibilidad del gas compensada con la temperatura (αt), y esta, a su vez, se puede obtener mediante un coeficiente CF. La librería de calibración del sensor contiene una tabla *lookup* con los valores de compensación (desde los $-10\text{ }^{\circ}\text{C}$ hasta los $50\text{ }^{\circ}\text{C}$) incluidos en la ficha técnica [35] para facilitar el cálculo.

```
static const float TEMP_COMPENSATION[61]={0.6745, 0.6854, 0.6963,  
0.7072, 0.7181, 0.7290, 0.7399, 0.7508, 0.7617, 0.7726, 0.7835, 0.7944,  
0.8053, 0.8162, 0.8271, 0.8380, 0.8489, 0.8598, 0.8707, 0.8816, 0.8925,  
0.9034, 0.9143, 0.9252, 0.9361, 0.9470, 0.9579, 0.9688, 0.9797, 0.9906,  
1.0000, 1.0075, 1.0150, 1.0225, 1.0300, 1.0375, 1.0450, 1.0525, 1.0600,  
1.0675, 1.0750, 1.0825, 1.0900, 1.0975, 1.1050, 1.1125, 1.1200, 1.1275,  
1.1350, 1.1425, 1.1500, 1.1575, 1.1650, 1.1725, 1.1800, 1.1875, 1.1950,  
1.2025, 1.2100, 1.2175, 1.2250};
```

Por tanto, si el valor medido de T^a está entre los incluidos en la tabla, se utiliza su coeficiente específico (para valores mayores se utiliza el máximo incluido, $50\text{ }^{\circ}\text{C}$, y para menores el mínimo, $-10\text{ }^{\circ}\text{C}$. Para el uso que se le va a dar al dispositivo no se precisa de valores más extremos).

La programación se ha realizado en la plataforma Keil μ Vision, un entorno de desarrollo y *debugger* de la compañía Arm, pero para el caso hay varias otras opciones válidas, como podría ser STM32CubeIDE.

Para leer los mensajes por el puerto serie, se conecta la placa STM32L053R8, con la placa de expansión para medir el gas acoplada, mediante USB a un PC y se utiliza el modo de depuración para ejecutar el código. Se escribirán las medidas de gas, la sensibilidad y la ganancia como expresiones de tipo `printf`, que se podrán leer abriendo una ventana desde el menú: *view* \rightarrow *Serial Windows* \rightarrow *Debug (printf) viewer*.



Figura 58. Salida del debugger de Keil para una medición de CO en un ambiente sin presencia de este gas

Se comprueba que en un ambiente exento de monóxido de carbono el sensor devuelve el valor correspondiente.

6.1.2 Integración de la conectividad wifi

En este apartado se va a especificar cómo se realiza la conexión entre el módulo wifi ESP-01s, basado en el chip ESP8266, y la placa SMT32 escogida para proporcionar conectividad wifi al dispositivo.

6.1.2.1 Alimentación ESP01-s

Para la alimentación del módulo ESP hay que tener en cuenta dos aspectos: la tensión de alimentación y sus requerimientos de corriente. Respecto a lo primero, el ESP-01s funciona a 3,3 V y admite un máximo de 3,6 V. En cuanto a la intensidad, se ha de consultar el *datasheet* del chip [36]. En él, se especifica que, alimentado a 3,3 V, el módulo requiere de más de 120 mA para la transmisión de datos y de más de 50 mA para la recepción.

Modes	Min	Typ	Max	Unit
Tx 802.11b, CCK 11 Mbps, POUT = +17 dBm	-	170	-	mA
Tx 802.11g, OFDM 54 Mbps, POUT = +15 dBm	-	140	-	mA
Tx 802.11n, MCS7, POUT = +13 dBm	-	120	-	mA
Rx 802.11b, 1024 bytes packet length , -80 dBm	-	50	-	mA
Rx 802.11g, 1024 bytes packet length , -70 dBm	-	56	-	mA
Rx 802.11n, 1024 bytes packet length , 65 dBm	-	56	-	mA
Modem-Sleep ^①	-	15	-	mA
Light-Sleep ^②	-	0.9	-	mA
Deep-Sleep ^③	-	10	-	μA
Power Off	-	0.5	-	μA

Figura 59. Consumo de corriente del chip ESP8266EX en sus diferentes modos de operación [36]

Teniendo en cuenta estos valores, se ha de determinar si es posible alimentar el módulo de forma estable mediante los pines de salida de la placa o es necesario algún elemento externo. En general, para hacer frente a los posibles picos de consumo en el envío de datos por wifi, se recomienda una intensidad mayor de 200 mA.

Symbol	Ratings	Max.	Unit
$\Sigma I_{VDD}^{(2)}$	Total current into sum of all V_{DD} power lines (source) ⁽¹⁾	105	mA
$\Sigma I_{VSS}^{(2)}$	Total current out of sum of all V_{SS} ground lines (sink) ⁽¹⁾	105	
ΣI_{VDD_USB}	Total current into V_{DD_USB} power lines (source)	25	
$I_{VDD(PIN)}$	Maximum current into each V_{DD} power pin (source) ⁽¹⁾	100	
$I_{VSS(PIN)}$	Maximum current out of each V_{SS} ground pin (sink) ⁽¹⁾	100	
I_{IO}	Output current sunk by any I/O and control pin except FTf pins	16	
	Output current sunk by FTf pins	22	
	Output current sourced by any I/O and control pin	-16	
$\Sigma I_{IO(PIN)}$	Total output current sunk by sum of all IOs and control pins except PA11 and PA12 ⁽²⁾	90	
	Total output current sunk by PA11 and PA12	25	
	Total output current sourced by sum of all IOs and control pins ⁽²⁾	-90	

Figura 60. Características de corriente de la placa STM32L053R8 [23]. Subrayado el valor de salida de corriente de los pines de la placa.

Al examinar la hoja técnica de la placa STM32L053R8 [23] se comprueba que, a pesar de que el voltaje de salida puede funcionar con lógica TTL 3,3 V, la corriente de salida se queda corta, especialmente si se utilizan las funcionalidades wifi, como se puede leer en la Figura 60. Por tanto, se determina que no es una opción viable utilizarla como fuente de alimentación del módulo ESP.

Para hacer frente al problema de la alimentación, por tanto, se va a utilizar un módulo de fuente de alimentación de 3,3 V y 5 V, un adaptador convertidor AC/DC de 9 V, 1 A y una *protoboard* para realizar el conexionado de pines.



Figura 61. Adaptador convertidor DC 9V 1A ³²

El módulo de fuente alimentación es un componente que reduce el voltaje de entrada (entre 6,5 V y 9 V, en el caso del que se va a utilizar) a uno con el que suelen funcionar los dispositivos electrónicos más pequeños: 3,3 V o 5 V.

Una de sus principales ventajas es que está diseñado para ser colocado directamente sobre una placa de prototipado y que cuenta con dos switches para elegir la tensión de cada uno de los buses de alimentación de la *protoboard*. La entrada de voltaje se realiza mediante un conector Jack de 5.5 mm x 2.1 mm y cuenta además con un conector USB de salida a 5V y un pulsador de corte.

³² Extraída de:
https://www.amazon.es/gp/product/B00VL1KQ3I/ref=ppx_yo_dt_b_asin_title_o00_s00?ie=UTF8&psc=1

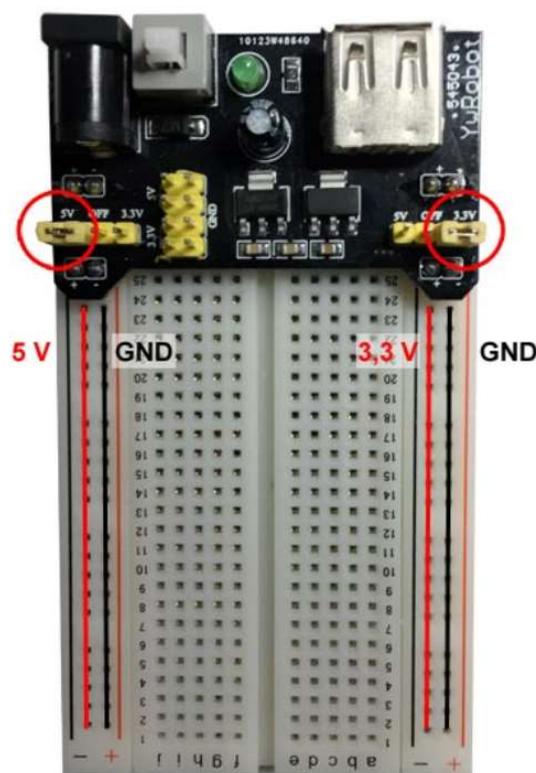


Figura 62. Módulo de fuente de alimentación de 3,3 V y 5 V conectado a una protoboard ³³

En definitiva, como la combinación entre el módulo y el adaptador de corriente proporcionan la tensión de 3,3 V requerida y una salida de corriente de hasta 700 mA, se considera una solución adecuada para suplir las necesidades eléctricas del módulo ESP01-s.

6.1.2.2 Pruebas previas transmisión

Previo a la conexión entre el módulo y la placa se puede realizar una prueba simple de los comandos para chequear su funcionamiento. Para ello, se utilizó en este caso el propio módulo ESP-01s, un módulo USB-TTL CH340G y un pc.

El módulo conversor USB-TTL convierte uno de los puertos USB de un ordenador en un puerto serie UART, con niveles de voltaje TTL (*transistor-transistor logic*, lógica transistor a transistor) compatibles con el ESP. Básicamente permite la comunicación entre los pines TX y RX del módulo y la entrada USB del pc utilizando el protocolo UART. En el mercado existen multitud de adaptadores de este tipo, en este caso se utilizó uno que incluía el chip CH340G y 5 pines: 2 de alimentación (3,3 y 5 V), TX, RX y masa.

³³ Extraída de: <https://programarfacil.com/podcast/como-configurar-esp01-wifi-esp8266/>



Figura 63. Módulo USB-TTL CH340G y cables Dupont ³⁴

En cuanto al módulo ESP-01s, como se ha mencionado anteriormente, dispone de 8 pines de conexión, de los cuales se usarán 6. La conexión entre este, el módulo de alimentación y el USB-TTL va del siguiente modo:

- Pin GND del ESP conectado a masa y al Pin GND del módulo USB-TTL.
- GPIO0 y GPIO2 al aire, no utilizados.
- Pin RX del ESP para la recepción de datos conectado al pin TX del módulo USB-TTL.
- Pin TX del ESP para la transmisión de datos conectado al pin RX del módulo USB-TTL.
- EN (*Enable*), Vcc y RESET del ESP al bus de 3,3 V de la *protoboard*.

Una vez se han conectado los pines y el módulo TTL a una entrada USB de un pc, es preciso utilizar una herramienta para poder leer la comunicación entre los dos dispositivos, así como escribir los comandos AT que se quieren probar. Hay diversos programas que simulan la comunicación por terminal en forma de cliente y que permiten transferir y leer datos con varios protocolos distintos, entre ellos PuTTY y Tera Term, el cual se va a utilizar en este caso.

Tera Term es un proyecto de código abierto de emulación de terminales que se puede descargar de forma gratuita. Este permite la comunicación entre servidor y cliente a partir del conjunto de protocolos TCP/IP, como Telnet y SSH, y la comunicación serie entre dos dispositivos. Esta última, definida en el apartado 5.2.1.1 sobre la comunicación UART/USART, es la que se va a utilizar para comunicar el pc y el módulo wifi.

³⁴ Extraída de: <https://www.tecnoteca.es/producto/schmod108-covertidor-de-usb-a-ttl-uart-c-i-ch340g/>



Figura 64. Configuración nueva conexión Tera Term

Al conectar el módulo USB-TTL al pc, nos aparece en la lista de puertos accesibles del programa el que se está utilizando en este caso para la comunicación. Si se selecciona, comienza la comunicación serie entre los dos dispositivos.

Desde *Setup* → *Serial port* se han de configurar los parámetros para la comunicación. Al ser asíncrona, se ha de ajustar velocidad de envío y recepción de datos para mantener una correcta transmisión. Puesto que el ESP8266 viene de fábrica con una velocidad de transferencia de señal de **115200 baudios** [25], se va a seleccionar esta misma velocidad en el programa. Además, la información se enviará en grupos de 8 bits, con un bit de parada y sin bit de paridad, puesto que así viene configurado el chip wifi.



Figura 65. Configuración y conexión del puerto serie Tera Term

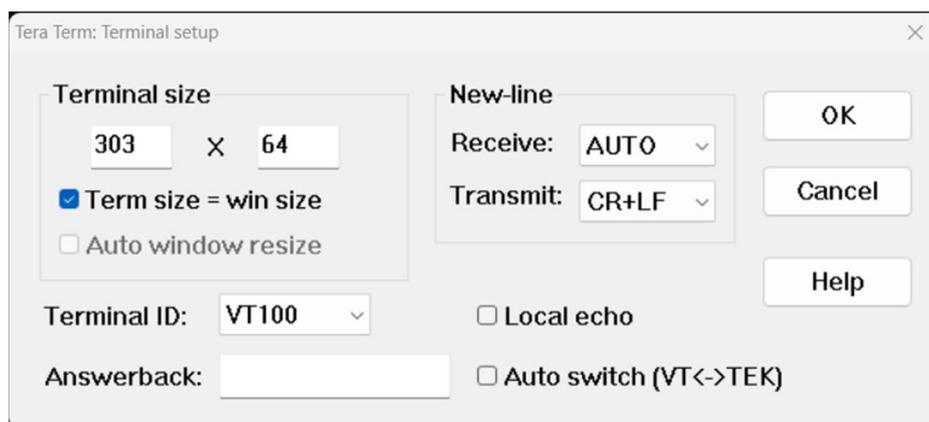


Figura 66. Configuración terminal Tera Term

Una vez configurada la comunicación, se pueden escribir distintos comandos por teclado en el terminal para probar si se envían correctamente al módulo wifi y si se obtiene respuesta. Entre los muchos incluidos en el firmware, a continuación se muestran algunos de ellos:

- AT: prueba estándar para comprobar si el sistema está funcionando correctamente.
- AT+RST: Reinicia el ESP8266.

Con esta prueba rápida se puede determinar que el módulo ESP-01s con el que se está trabajando funciona correctamente y, además, se aprovecha el cableado realizado para el resto del proyecto.

6.1.2.3 Integración hardware módulo wifi y microcontrolador STM32L0

Con el objetivo de otorgar funcionalidades inalámbricas al microcontrolador, se ha determinado conectarlo al chip wifi ESP8266. A fin de transferir los datos precisos del MCU a dicho chip, se va a configurar una comunicación por protocolo UART, anteriormente definida.

Para que la comunicación por protocolo UART se pueda producir entre los dos dispositivos, se debe conectar uno de los pines de transmisión (TX) de la placa NUCLEO-L053R8 con el pin de recepción (RX) del módulo wifi y uno de los pines de recepción de la placa con el de transmisión del módulo. Además, es necesario conectar las masas entre ellas para que funcione correctamente.

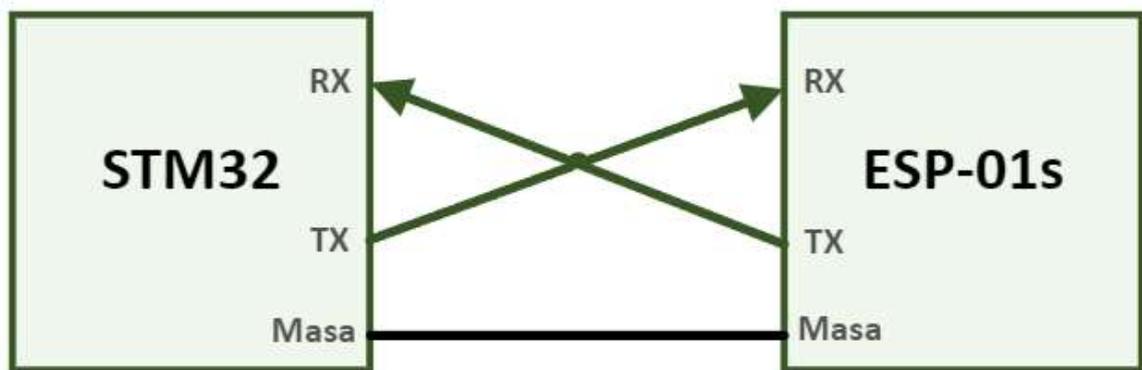


Figura 70. Ilustración de la comunicación USART/UART entre el microcontrolador STM32L0 y el módulo wifi ESP-01s (elaboración propia)

Se ha de revisar la documentación de la *board* escogida para determinar cuáles de sus pines exteriores están conectados a patillas del MCU que soportan la comunicación UART/USART. Como hay varios, en este caso se van a seleccionar los pines A14 y A15 (Figura 71).

Port	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7	
	SPI1/TIM21/SYS_A F/EVENTOUT/	LCD	USB/TIM2/ EVENTOUT/	TSC/ EVENTOUT	USART1/2/3	TIM2/21/22	EVENTOUT	COMP1/2	
Port A	PA0	-	-	TIM2_CH1	TSC_G1_IO1	USART2_CTS	TIM2_ETR	-	COMP1_OUT
	PA1	EVENTOUT	LCD_SEG0	TIM2_CH2	TSC_G1_IO2	USART2_RTS/ USART2_DE	TIM21_ETR	-	-
	PA2	TIM21_CH1	LCD_SEG1	TIM2_CH3	TSC_G1_IO3	USART2_TX	-	-	COMP2_OUT
	PA3	TIM21_CH2	LCD_SEG2	TIM2_CH4	TSC_G1_IO4	USART2_RX	-	-	-
	PA4 (1)	SPI1_NSS	-	-	TSC_G2_IO1	USART2_CK	TIM22_ETR	-	-
	PA5	SPI1_SCK	-	TIM2_ETR	TSC_G2_IO2	-	TIM2_CH1	-	-
	PA6	SPI1_MISO	LCD_SEG3	-	TSC_G2_IO3	LPUART1_CTS	TIM22_CH1	EVENTOUT	COMP1_OUT
	PA7	SPI1_MOSI	LCD_SEG4	-	TSC_G2_IO4	-	TIM22_CH2	EVENTOUT	COMP2_OUT
	PA8	MCO	LCD_COM0	USB_CRSS_SYNC	EVENTOUT	USART1_CK	-	-	-
	PA9	MCO	LCD_COM1	-	TSC_G4_IO1	USART1_TX	-	-	-
	PA10	-	LCD_COM2	-	TSC_G4_IO2	USART1_RX	-	-	-
	PA11	SPI1_MISO	-	EVENTOUT	TSC_G4_IO3	USART1_CTS	-	-	COMP1_OUT
	PA12	SPI1_MOSI	-	EVENTOUT	TSC_G4_IO4	USART1_RTS/ USART1_DE	-	-	COMP2_OUT
	PA13	SWDIO	-	USB_NOE	-	-	-	-	-
	PA14	SWCLK	-	-	-	USART2_TX	-	-	-
PA15	SPI1_NSS	LCD_SEG17	TIM2_ETR	EVENTOUT	USART2_RX	TIM2_CH1	-	-	

Figura 71. Tabla de las funciones alternas de los pines GPIO del puerto A del microcontrolador STM32L0 [23]

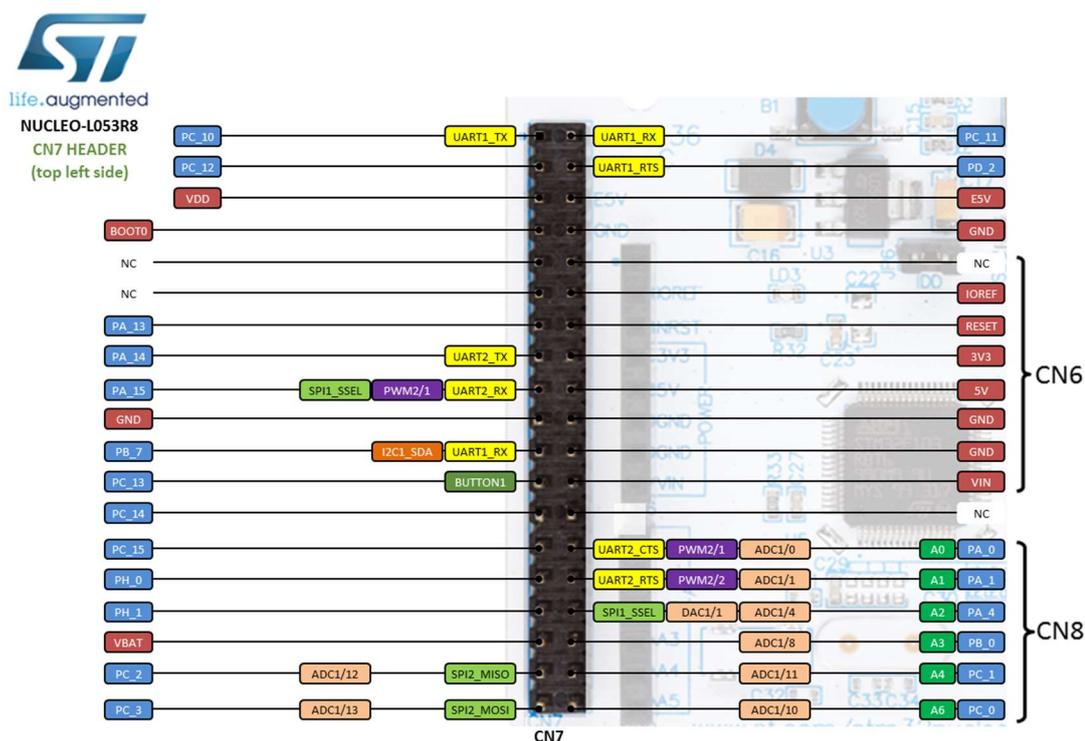


Figura 72. STM32L053R8 Morpho header CN7 ³⁵

³⁵ Extraída de: <https://os.mbed.com/platforms/ST-Nucleo-L053R8/>

En cuanto al cableado para la alimentación eléctrica del micro y la conexión con el módulo ESP-01s, se va a trabajar partiendo de lo explicado en la sección sobre la alimentación del ESP01-s anterior, es decir, se van a conectar algunos de los *headers* hembra y pines tipo Arduino de la placa a la *protoboard* alimentada por el módulo convertidor AC/DC de 9 V.

Operating voltage	1.8 V to 3.6 V (down to 1.65 V at power-down) with BOR option 1.65 V to 3.6 V without BOR option
--------------------------	---

Figura 73. Rango tensión alimentación STM32L0 [23]

El rango de tensión de alimentación eléctrica del STM32 oscila entre 1,8 V y 3,6 V (Figura 73). Para suministrarlos, se puede utilizar alguno de los pines de entrada de tensión o bien la entrada micro-USB que también se usa para escribir el programa en la memoria. En este caso se ha seleccionado el pin tipo Arduino marcado como 3,3 V y se ha conectado a la línea de la placa de prototipado que suministra ese mismo voltaje. La masa de la placa, uno de los pines GND en la misma fila, se ha conectado a la línea de masa de la *protoboard* adyacente.

Resumen conexión pines STM32 y ESP01s		
Pin STM32	Pin ESP-01s	Protoboard
A14 (TX)	RX	Columna 27
A15 (RX)	TX	Columna 29
3V3	-	3,3 V
GND	GND	GND
-	Vcc	3,3 V
-	EN	3,3 V
-	RST	3,3 V

Tabla 15: Resumen conexión entre la board y el módulo wifi

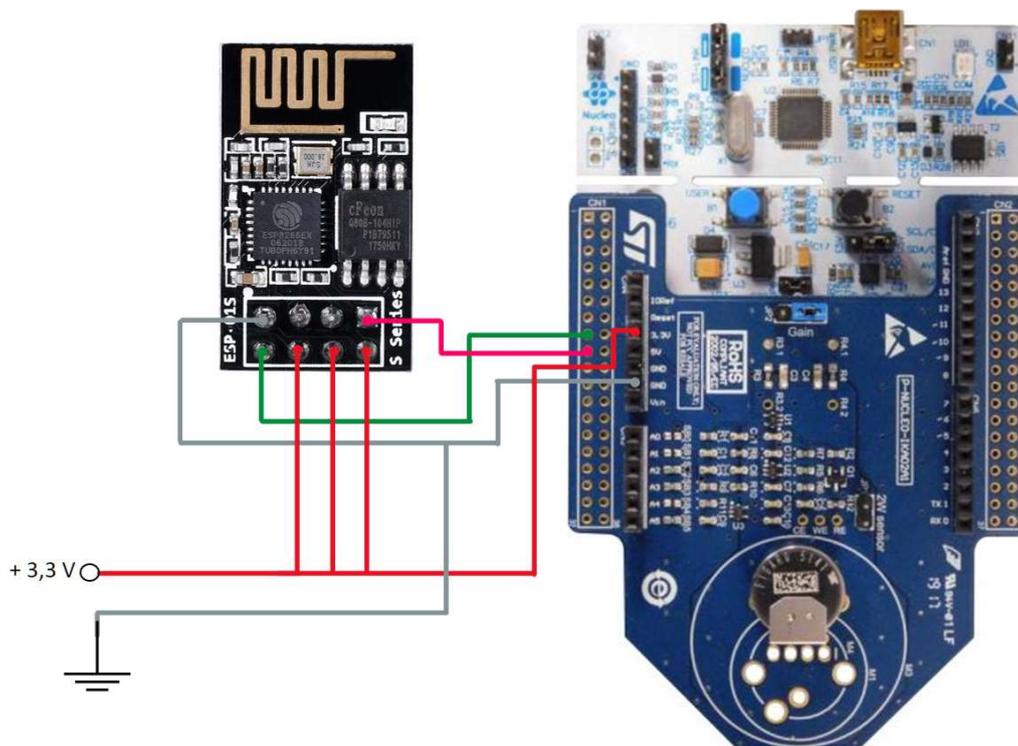


Figura 74. Ilustración conexasión del módulo wifi ESP01s, placa STM32L053R8 y módulo de expansión P-NUCLEO-IKA02A1 (elaboración propia)

6.1.3 Elaboración maqueta ventilador

6.1.3.1 Controlador y motor

Ya mencionado anteriormente, los pines de salida de la *board* funcionan con lógica de 3,3 V TLL y una corriente de salida de 16 mA. Estos valores no son suficientes para iniciar la rotación del motor Nidec DMN29BA seleccionado, el cual, de acuerdo con su ficha de datos [20], funciona a 12 V con una corriente nominal de 420 mA (70 mA sin carga).

TYPE	RATED						NO LOAD		STALL		WEIGHT	
	OUT PUT	VOLTAGE	TORQUE		CURRENT	SPEED	CURRENT	SPEED	TORQUE		g	lb
	W	V	mN·m	oz·in	A	r/min	A	r/min	mN·m	oz·in		
DMN29BA	3.0	12	7.8	1.11	0.42	3700	0.07	5000	30	4.17	90	0.20
DMN29BB	3.0	24	7.8	1.11	0.21	3700	0.05	5000	30	4.17	90	0.20

Figura 75. Resumen características motor Nidec DMN29BA [16]

Esta es la razón principal para utilizar el módulo X-Nucleo-IHM13A1, que incluye un *driver* para el control de motores DC con escobillas.

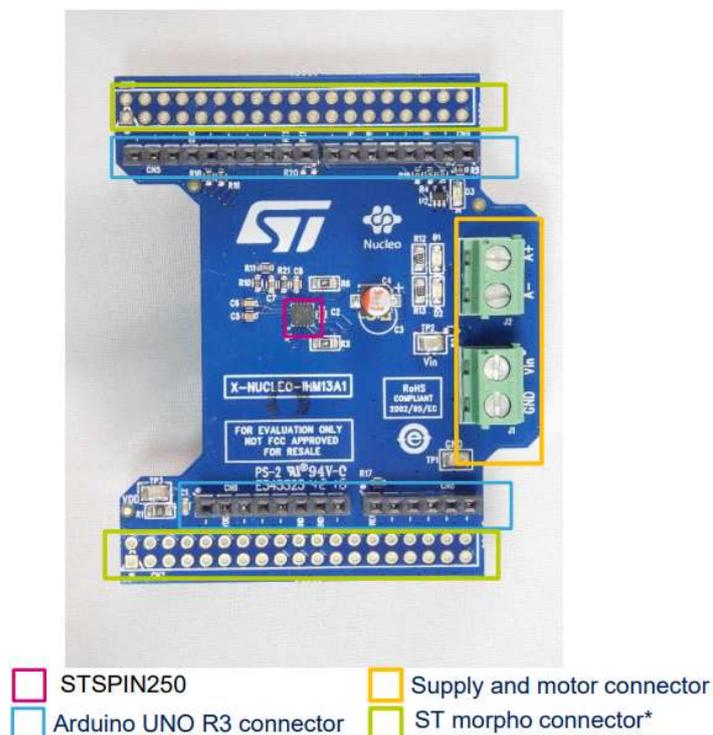


Figura 76. Visión general de las partes y componentes de la placa X-Nucleo-IHM13A1 ³⁶

Este módulo está pensado para ser conectado directamente, entre otras, sobre la placa de desarrollo STM32L053R8 ya que tiene su misma configuración de conectores Arduino y ST *morpho*, así que por simplicidad y facilidad de conexión esto es lo que se hará. Asimismo, tiene dos pares de conectores de externos para alimentar el controlador y el motor.

En el esquema del circuito de la placa X-NUCLEO-IHM13A1 (Figura 77), obtenido de su ficha técnica [33], se pueden ver las conexiones del controlador. Hay que asegurarse que los pines que necesita para funcionar no se utilizan para ninguna otra función.

³⁶ Extraída de: https://www.st.com/resource/en/product_presentation/x-nucleo-ihm13a1_quick_start_guide.pdf

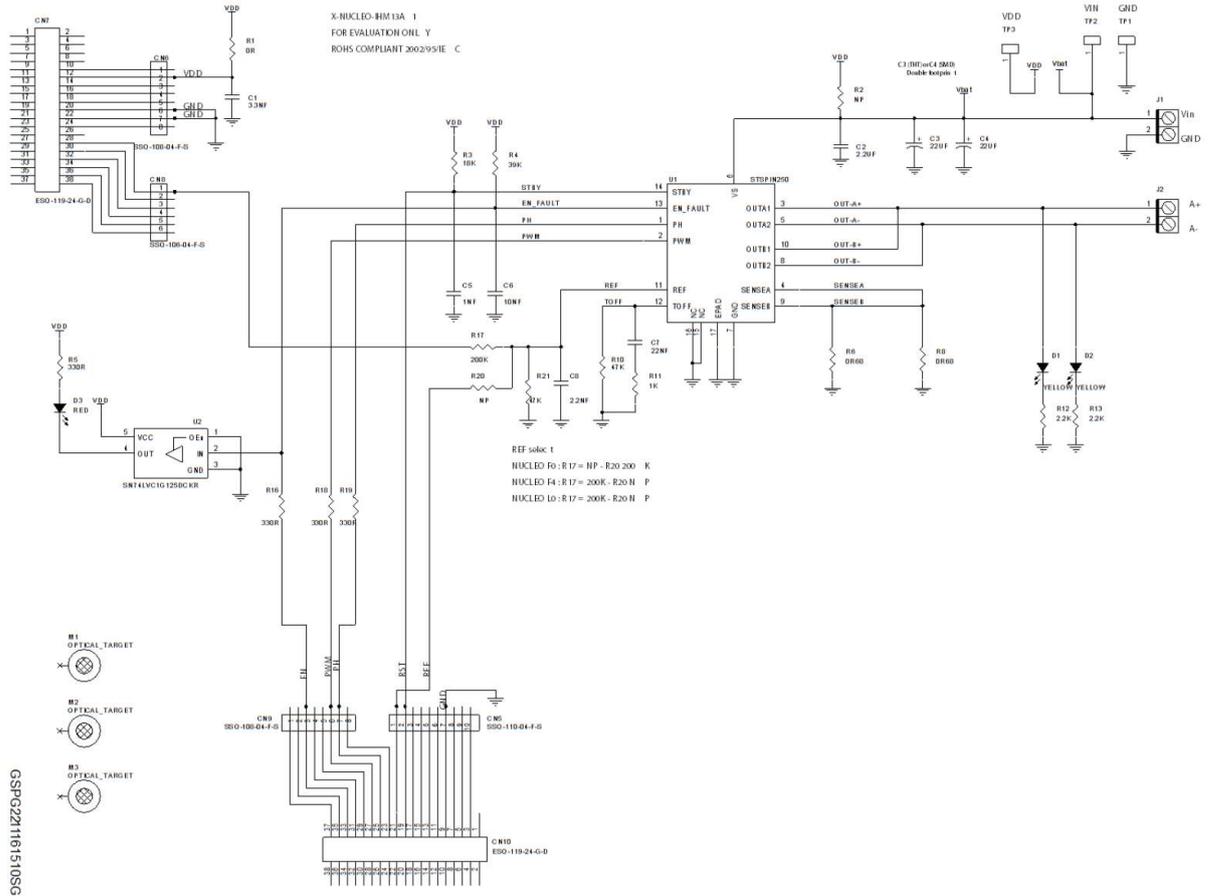


Figura 77. Esquema del circuito de la placa X-NUCLEO-IHM13A1 [33]

Para realizar el control de dicho motor por modulación de ancho de pulso se precisa de conectar el *driver* STSPIN250 con una de las salidas de uno de los *timers* de la placa, que son los que generan este tipo de señales. Según su hoja de datos [23], esta incluye tres de propósito general, uno de bajo consumo y otro básico, así como dos con función de perro guardián (*watchdog*) y el *SysTick timer*, común en los procesadores Cortex para generar interrupciones.

Peripheral		STM32L053C6	STM32L053R6	STM32L053C8	STM32L053R8
Flash (Kbytes)			32		64
Data EEPROM (Kbytes)			2		2
RAM (Kbytes)			8		8
Timers	General-purpose		3		3
	Basic		1		1
	LPTIMER		1		1

Figura 78. STM32L053R8 características y periféricos. Subrayados los timers de la placa.

La entrada PWM del controlador STSPIN250 viene por construcción conectada al sexto pin del canal 9 del módulo X-NUCLEO-IHM13A1 (Figura 77). Puesto que se va a conectar este módulo sobre la placa STM32L053R8, las conexiones coincidirán. Según su asignación de pines (Figura 79), al acoplar el módulo, la entrada PWM del *driver* acabará conectada al pin B4, en el cual se puede asignar la salida del canal 1 del TIM22 (Figura 79 y Figura 80).

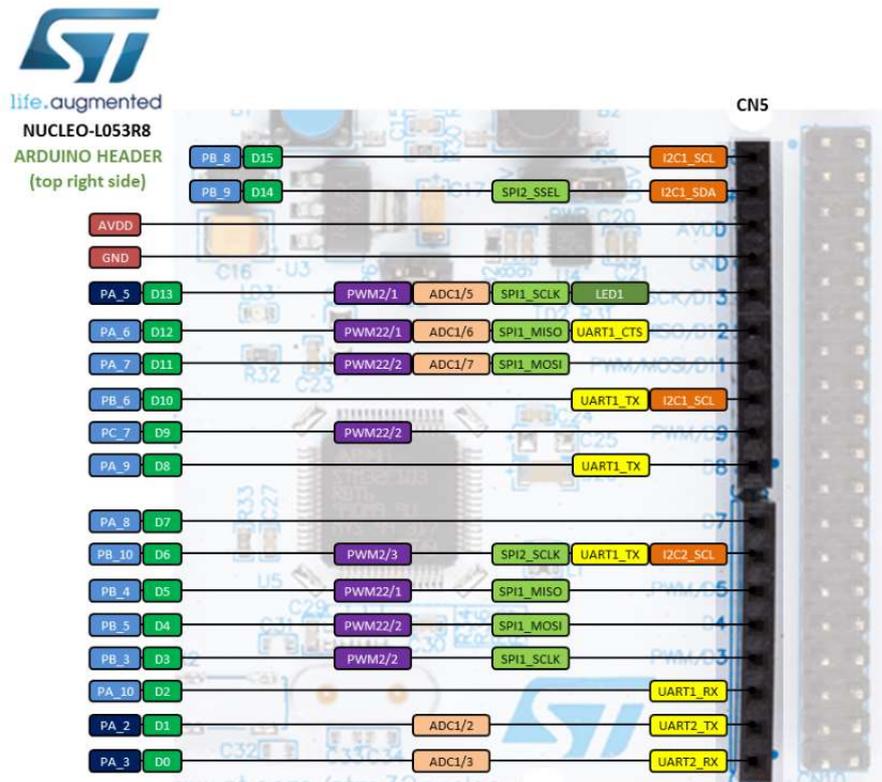


Figura 79. STM32L053R8 Arduino header CN9 ³⁷

³⁷ Extraída de: <https://os.mbed.com/platforms/ST-Nucleo-L053R8/>

Port	AF0	AF1	AF2	AF3	AF4	AF5	AF6	
	SPI1/SPI2/I2S2/ USART1/ EVENTOUT/	I2C1/LCD	LPUART1/LPTIM /TIM2/SYS_AF/ EVENTOUT	I2C1/TSC	I2C1/TIM22/ EVENTOUT/ LPUART1	SPI2/I2S2/I2C2	I2C2/TIM21/ EVENTOUT	
Port B	PB0	EVENTOUT	LCD_SEG5	-	TSC_G3_IO2	-	-	
	PB1	-	LCD_SEG6	-	TSC_G3_IO3	LPUART1_RTS/ LPUART1_DE	-	
	PB2	-	-	LPTIM1_OUT	TSC_G3_IO4	-	-	
	PB3	SPI1_SCK	LCD_SEG7	TIM2_CH2	TSC_G5_I01	EVENTOUT	-	
	PB4	SPI1_MISO	LCD_SEG8	EVENTOUT	TSC_G5_IO2	TIM22_CH1	-	
	PB5	SPI1_MOSI	LCD_SEG9	LPTIM1_IN1	I2C1_SMBA	TIM22_CH2	-	
	PB6	USART1_TX	I2C1_SCL	LPTIM1_ETR	TSC_G5_IO3	-	-	
	PB7	USART1_RX	I2C1_SDA	LPTIM1_IN2	TSC_G5_IO4	-	-	
	PB8	-	LCD_SEG16	-	TSC_SYNC	I2C1_SCL	-	
	PB9	-	LCD_COM3	EVENTOUT	-	I2C1_SDA	SPI2_NSS/I2S2_ WS	
	PB10	-	LCD_SEG10	TIM2_CH3	TSC_SYNC	LPUART1_TX	SPI2_SCK	I2C2_SCL
	PB11	EVENTOUT	LCD_SEG11	TIM2_CH4	TSC_G6_IO1	LPUART1_RX	-	I2C2_SDA
	PB12	SPI2_NSS/I2S2_WS	LCD_SEG12	LPUART1_RTS/ LPUART1_DE	TSC_G6_IO2	-	I2C2_SMBA	EVENTOUT
	PB13	SPI2_SCK/I2S2_CK	LCD_SEG13	-	TSC_G6_IO3	LPUART1_CTS	I2C2_SCL	TIM21_CH1
	PB14	SPI2_MISO/I2S2_MCK	LCD_SEG14	RTC_OUT	TSC_G6_IO4	LPUART1_RTS/ LPUART1_DE	I2C2_SDA	TIM21_CH2
PB15	SPI2_MOSI/I2S2_SD	LCD_SEG15	RTC_REFIN	-	-	-	-	

Figura 80. Tabla de las funciones alternas de los pines GPIO del puerto B del microcontrolador STM32L0 [23]

El TIM22 es un temporizador de propósito general basado en un contador de 16 bits incremental/decremental. Sus características se pueden encontrar en la tabla de la Figura 81, entre las que destacan un *prescaler* de 16 bits para dividir y ajustar la frecuencia del reloj interno y la existencia de dos canales independientes que se pueden usar para generar pulsos, señales PWM o como entrada/salida de captura comparativa para poder medir simultáneamente varios eventos.

Timer	Counter resolution	Counter type	Prescaler factor	DMA request generation	Capture/compare channels	Complementary outputs
TIM2	16-bit	Up, down, up/down	Any integer between 1 and 65536	Yes	4	No
TIM21, TIM22	16-bit	Up, down, up/down	Any integer between 1 and 65536	No	2	No
TIM6	16-bit	Up	Any integer between 1 and 65536	Yes	0	No

Figura 81. Comparación timers STM32L053R8 [23]

El mencionado *prescaler* permite ajustar de forma más precisa la cuenta al dividir el valor reloj entre un número entre 1 y 65536 (Figura 81). Por ejemplo, para un reloj del sistema de 32 MHz, un *prescaler* con valor 32 dará lugar a una cuenta con una frecuencia de 1 MHz [37].

Para generar una señal PWM adecuada habrá que configurar ciertos parámetros por programa. La siguiente expresión representa la frecuencia de la señal con respecto a la

frecuencia del reloj, el *prescaler* con el que trabaja el *timer* y el valor de auto recarga (el valor hasta el cual el temporizador contará antes de reiniciarse) [37]. Se puede observar que para ambos valores se debe sumar una unidad para compensar que ambas variables empiezan por 0.

$$f_{PWM} = \frac{f_{TIM}}{(\text{prescaler} + 1) \cdot (\text{contador} + 1)} = \frac{f_{TIM}}{(PSC + 1) \cdot (ARR + 1)}$$

Para el control de motores DC por PWM, un valor adecuado de frecuencia suele rondar entre los 200 y los 20000 Hz, dependiendo de sus características y de las posibles vibraciones o ruidos. Si se consulta la hoja de datos del ventilador cuya etapa de control se está tratando de replicar [19], se puede comprobar que el ajuste de velocidad por PWM está diseñado para una señal de entre 1 y 10 kHz aplicada entre sus terminales 1 y 2 (Figura 82). Por tanto, se configurará la señal con una frecuencia de modulación de 8 kHz.

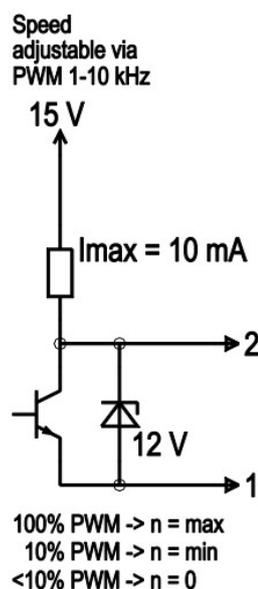


Figura 82. Circuito de control por PWM ventilador W3G300-UL11-35 [19]

La frecuencia de reloj del temporizador se podrá determinar por programa, escogiéndose trabajar a 32 MHz.

Con esto, solo queda determinar los dos valores de registro (auto recarga y *prescaler*) de modo que se obtenga una señal PWM con la mejor resolución posible. Para ello, se puede utilizar un proceso iterativo en el que se empieza con el segundo igualado a 0, es decir, sin división de reloj. Si el valor de auto registro acabara siendo superior a un valor de 16 bits, es decir, superior a 65536, habría que aumentar el valor de *prescaler* en 1 y repetir el proceso. Sin embargo, no es el caso para los valores asignados:

$$f_{PWM} = \frac{f_{TIM}}{(PSC + 1) \cdot (ARR + 1)}$$

$$ARR = \frac{f_{TIM}}{(PSC + 1) \cdot f_{PWM}} - 1$$

$$PSC = 0$$

$$ARR = \frac{f_{TIM}}{f_{PWM}} - 1 = \frac{32 \cdot 10^6}{8 \cdot 10^3} - 1 = 3999$$

Estos valores de registro serán los que luego habrá que escribir en una función para configurar la señal PWM del temporizador TIM22.

Luego, para modificar la velocidad a la que gira el motor, bastará con modificar su ciclo de trabajo. Esto se puede hacer cambiando el registro de comparación (CCR), que se utiliza para comparar el valor del contador del temporizador con un valor específico, realizando alguna acción cuando se alcanza o supera. En este caso, el CCR se usará para determinar el punto en el cual la señal de salida cambia de estado.

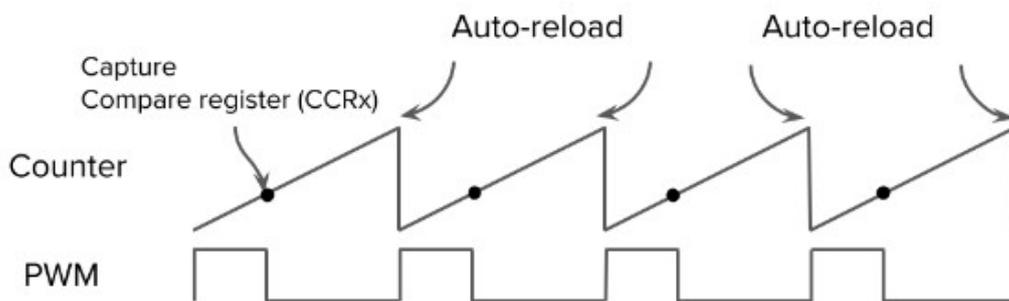


Figura 83. Generación de una señal PWM con la salida de un contador³⁸

Se puede aplicar la siguiente fórmula para modificar el *duty cycle* según los valores de registro y auto recarga:

$$\text{Ciclo de Trabajo} = \frac{CCRx + 1}{ARR + 1}$$

$$CCRx = (\text{Ciclo de Trabajo} \cdot (ARR + 1)) - 1$$

Para un ciclo de trabajo del 80%, por ejemplo, quedaría:

$$CCRx = (0,8 \cdot (3999 + 1)) - 1 = 3199$$

³⁸ Extraída de: <https://www.stepschool.com/pages/blog?p=stm32-pwm-stm32cubemx>

Que es el valor de registro de comparación que luego se pasaría al *timer*.

Por otro lado, la tensión de alimentación del controlador para suministrar la potencia necesaria para mover el motor según su hoja de datos ha de estar entre 1,8 y 10 V. Para ello, se utilizarán los conectores externos marcados como GND/Vin (pin 6 del STSPIN250) conectados al bus de 5 V DC de la *protoboard*. Los conectores adyacentes A-/A+ (pines 3 y 5 del STSPIN250) se usarán para alimentar el motor. El resultado de la conexión es similar a la que se puede ver en la Figura 84.

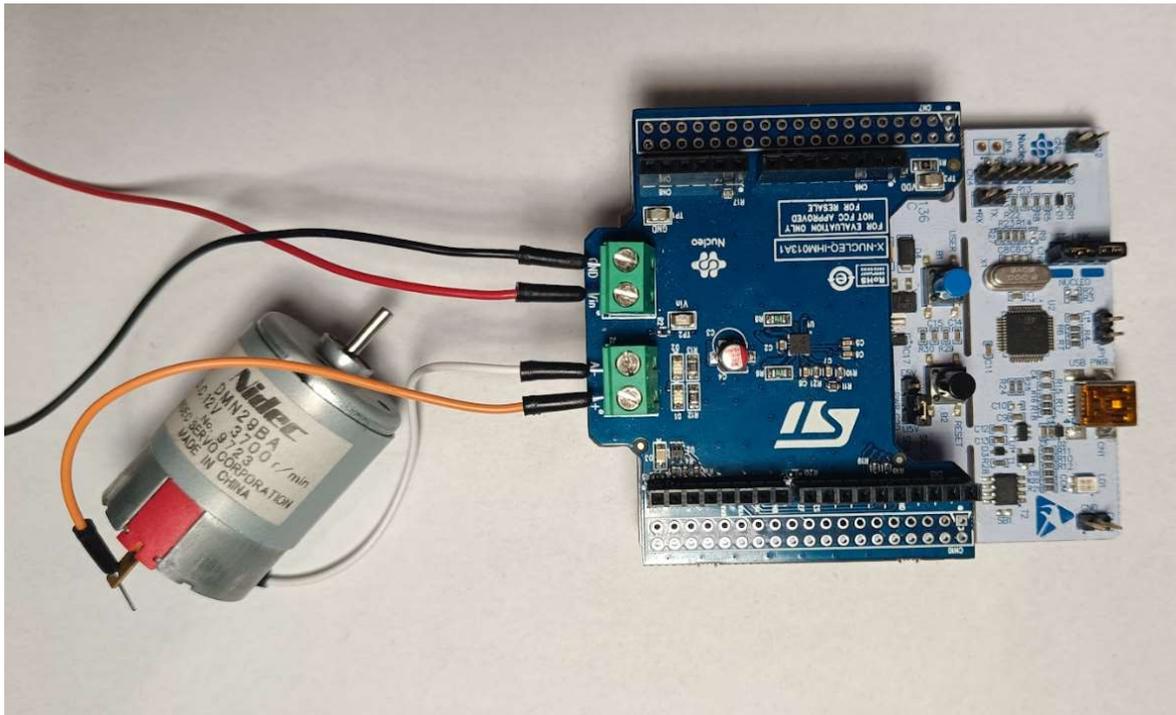


Figura 84. Placa X-NUCLEO-IHM13A1 conectada a una STM32 board para el control de un motor DC

6.1.3.2 Encoder

Ya se ha mencionado previamente que, para la longitud del eje del motor seleccionado, 12 mm [20], el AMT103 debería encajar sin problemas, puesto que requiere de mínimo de 9 mm para el montaje. Este es muy sencillo de realizar siguiendo paso a paso el manual o el vídeo proporcionados por el fabricante [38]. El DIP switch se dejará en la posición por defecto (resolución de 2048 PPR) y, para ajustar el engranaje de modo que gire junto al motor, se utilizará el manguito naranja para ejes de 3 mm.

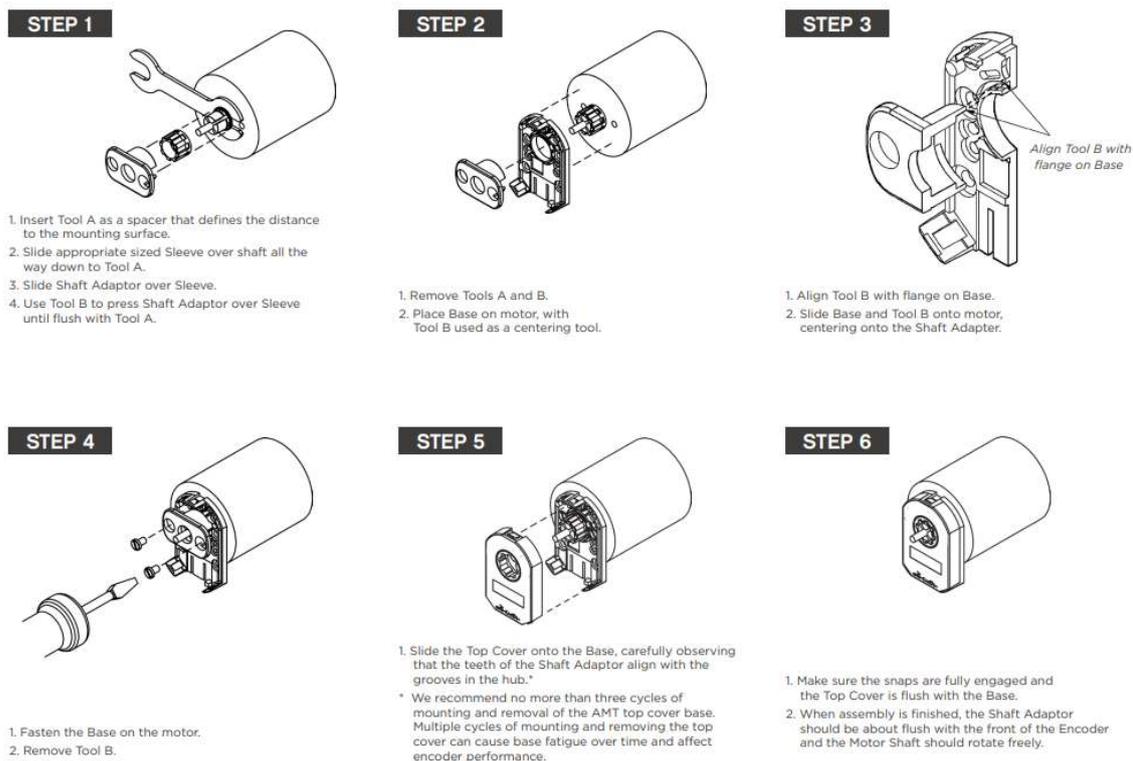


Figura 85. Manual de montaje del encoder AMT103 en un motor [38]

Teniendo en cuenta que su tensión mínima de suministro es de 3,6 V y la típica de 5 [39], no se obtendría suficiente voltaje alimentando este dispositivo con una de las salidas de la *board* STM32L053R8. Por tanto, puesto que se va a utilizar una de las salidas del módulo de fuente de alimentación para proporcionar 5 V a un bus de la placa de prototipado, se aprovechará para conectar un cable desde él al pin de 5 V del *encoder* y otro entre la masa adyacente y el pin G.

Por otro lado, aunque se dispone de dos pines de salida para las ondas del canal A y B, para el caso actual, en el que únicamente se va a medir la velocidad de giro en un sentido, se simplifica la conexión utilizando solo el pin índice (X). Este proporciona una señal de nivel alto cada vez que el eje completa un giro. Normalmente se utiliza como referencia absoluta de posición, pero, si se cuentan los pulsos de salida de este pin en cierto tiempo, el cálculo de velocidad es muy sencillo ($v (rpm) = \frac{n \text{ pulsos}}{T}$). Por ende, se conectará con un cable al pin C8 de la placa, que más adelante se programará para leer sus señales.

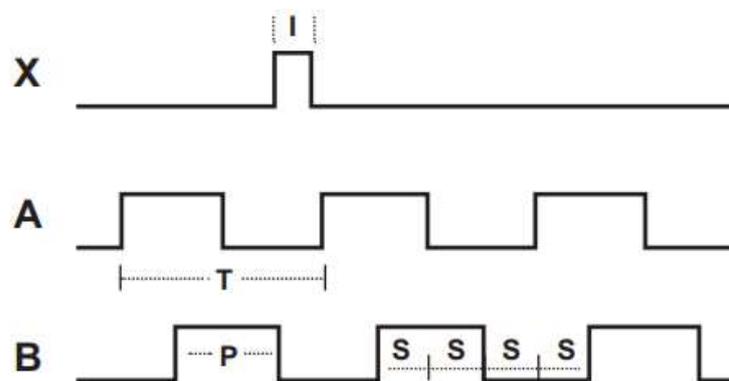


Figura 86. Señales de salida del codificador rotatorio [39]

6.1.4 Resumen montaje y conexiones

Para recapitular, se muestran en la Tabla 16 los pines utilizados de la placa de control y su función.

Resumen asignación de pines STML053R8	
Pin	Función
A2	Entrada analógica. Referencia de tensión para el sensor de CO
A3	Entrada analógica. Lectura tensión concentración de gas
A5	Entrada analógica. Lectura tensión temperatura ambiente
A10	Salida digital. Pin Enable del driver del motor
A14	Pin UART transmisión datos (TX)
A15	Pin UART recepción datos (RX)
B4	Señal PWM. Canal 1 TIM22
B10	Salida digital. Puerto A para el control de dirección driver del motor
C7	Salida digital. Pin Reset del driver del motor
C8	Entrada digital. Lectura del pin índice del encoder
C13	Entrada digital. Botón azul USER
3V3	Alimentación
GND	Masa

Tabla 16: Resumen de la asignación de los pines GPIO de la board STML053R8

Asimismo, al finalizar el montaje tal y como se indica en las secciones anteriores, el dispositivo al completo queda como en la Figura 87. Sus partes están definidas en la Tabla 17.

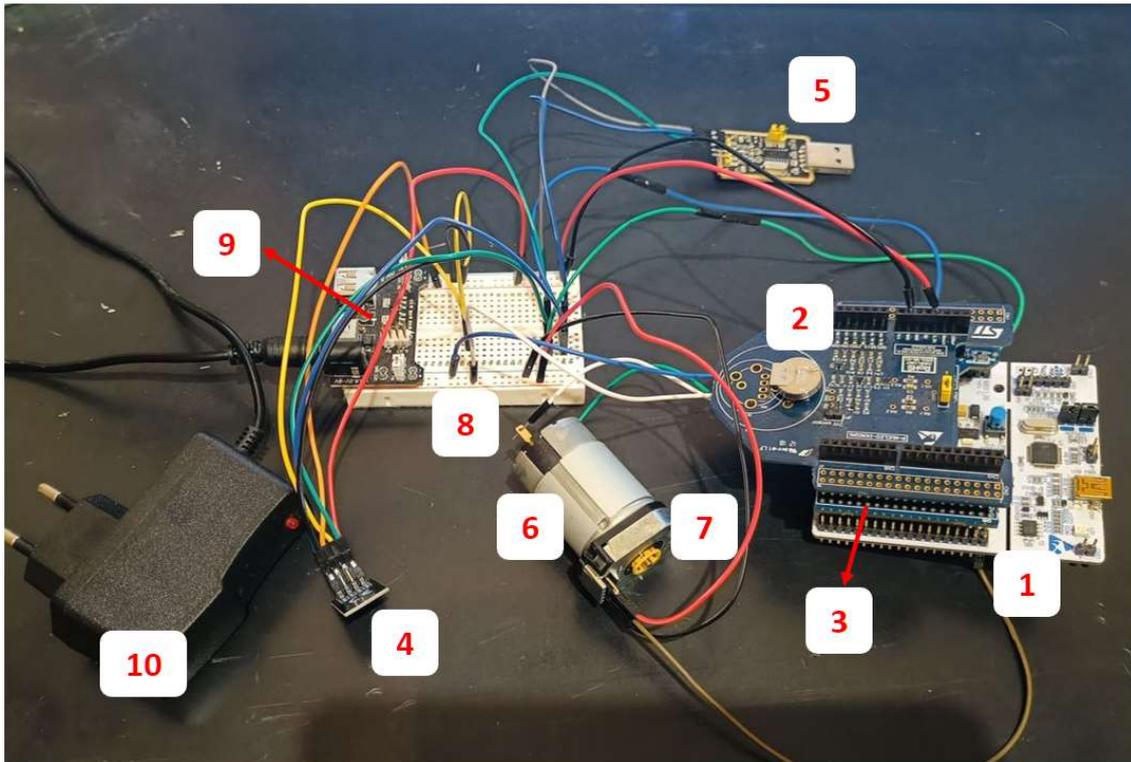


Figura 87. Dispositivo montado

Definición etiquetas Figura 87. Dispositivo montado	
Etiqueta	Componente
1	Placa de desarrollo STM32L053R8
2	Placa de expansión X-NUCLEO-IKA02A1
3	Placa de expansión X-NUCLEO-IHM13A1
4	Módulo wifi ESP-01S
5	UART-TTL USB Adaptador CH340G
6	Motor DC Nidec serie DMN29, 12 V
7	Codificador giratorio AMT10E3-V
8	Protoboard
9	Modulo para fuente de alimentación de 3.3 V – 5 V
10	Adaptador convertidor DC 9V

Tabla 17. Definición etiquetas Figura 87. Dispositivo montado

6.2 Desarrollo del software

En el apartado de software se resumen las operaciones realizadas para programar las distintas partes del proyecto (microcontrolador, interfaz gráfica y aplicación de visualización de datos) de acuerdo con sus funciones.

6.2.1 Programación MCU

6.2.1.1 STM32CubeMX

Para inicializar los pines deseados del microcontrolador de forma simple, se puede utilizar la herramienta gratuita STM32CubeMX, desarrollada por ST para configurar fácilmente hardware, periféricos y placas Núcleo.

Al iniciar el programa, es posible trabajar partiendo de un microcontrolador o placa en específico. En este caso se selecciona la *board* NUCLEO-L053R8.

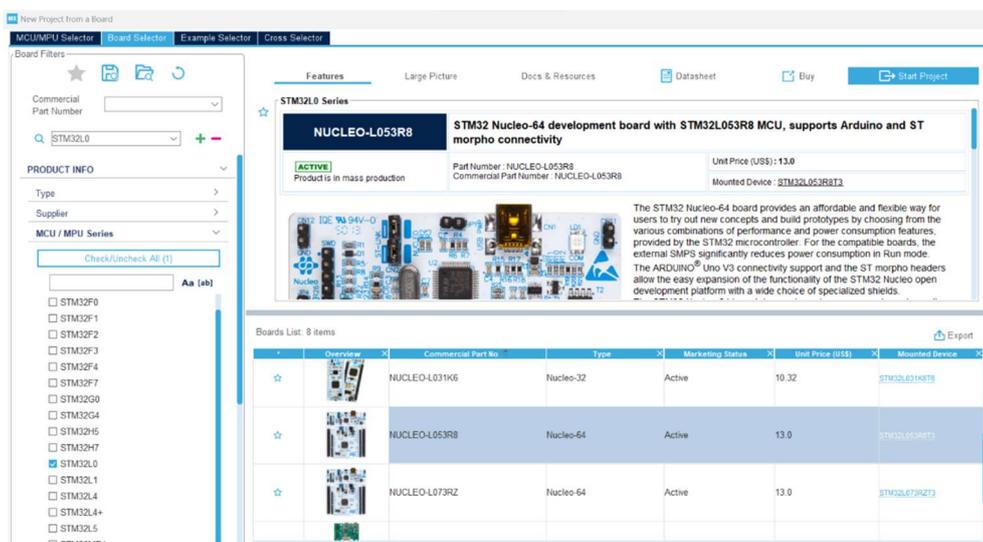


Figura 88. CubeMX selección de la placa

Inicializando dicha placa en su configuración estándar, obtenemos un esquemático del microcontrolador elegido. En él, se puede seleccionar cada patilla y escoger su modo de funcionamiento.

En la parte izquierda de la aplicación hay un listado de categorías para configurar los pines. Puesto que se va a utilizar la comunicación USART, seleccionamos los pines TX y RX elegidos y, desde el desplegable “Connectivity”, se escoge la opción USART2. Como se ha mencionado previamente, este micro cuenta con 2 interfaces USART, por lo que USART1 también sería válida (ambas tienen las mismas funcionalidades y la misma velocidad de hasta 4 Mbit/s).

USART modes/features ⁽¹⁾	USART1 and USART2
Hardware flow control for modem	X
Continuous communication using DMA	X
Multiprocessor communication	X
Synchronous mode ⁽²⁾	X
Smartcard mode	X
Single-wire half-duplex communication	X
IrDA SIR ENDEC block	X
LIN mode	X
Dual clock domain and wakeup from Stop mode	X
Receiver timeout interrupt	X
Modbus communication	X
Auto baud rate detection (4 modes)	X
Driver Enable	X

1. X = supported.
2. This mode allows using the USART as an SPI master.

Figura 89. Implementación tecnología USART en el microcontrolador STM32L0 [23]

Para la comunicación USART se escogen los pines A14 y A15 y se selecciona además el modo de comunicación asíncrono y, puesto que en ella los dos dispositivos han de transmitir a la misma velocidad, en "Parameter Settings" se asigna un *baud rate* de 115200 bit/s, el mismo que tiene preconfigurado el ESP8266. De igual forma, se ha de comprobar que los parámetros de longitud de palabra, bit de paridad y bit de parada coinciden con los del chip wifi (8, sin y 1, respectivamente) [36].

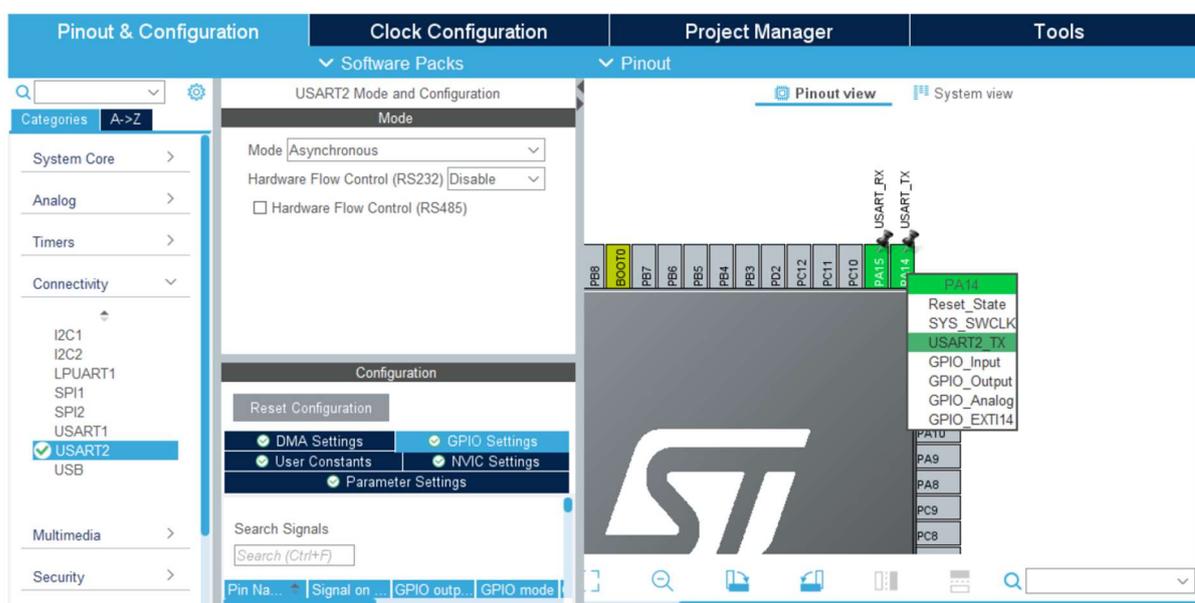


Figura 90. CubeMX selección pines USART



Figura 91. CubeMX parámetros pines UART

La configuración de los pines PA2, PA3 y PA5 como entradas analógicas para medir el voltaje de referencia, la temperatura y el monóxido de carbono, así como el ADC se realizará por programa a partir de la librería de ST.

Para la generación de la señal PWM se utilizará el canal 1 del *timer* TIM22 que, como se ha comentado, puede generarse desde el pin 4 del puerto B. En “*Timers*”, se selecciona dicho temporizador y, en la ventana con sus características, se marca por un lado el reloj interno como fuente de reloj y, por el otro, se selecciona “Generación de señal PWM CH1” en el canal 1.



Figura 92. CubeMX Selección pin Canal 1 TIM22

Para configurar otro *timer*, el 6, que se utilizará para regular una tarea periódica, también se escoge desde “*Timers*” y se habilita. En propiedades, se pueden ajustar los valores de cuenta y prescaler, pero se hará luego por programación.

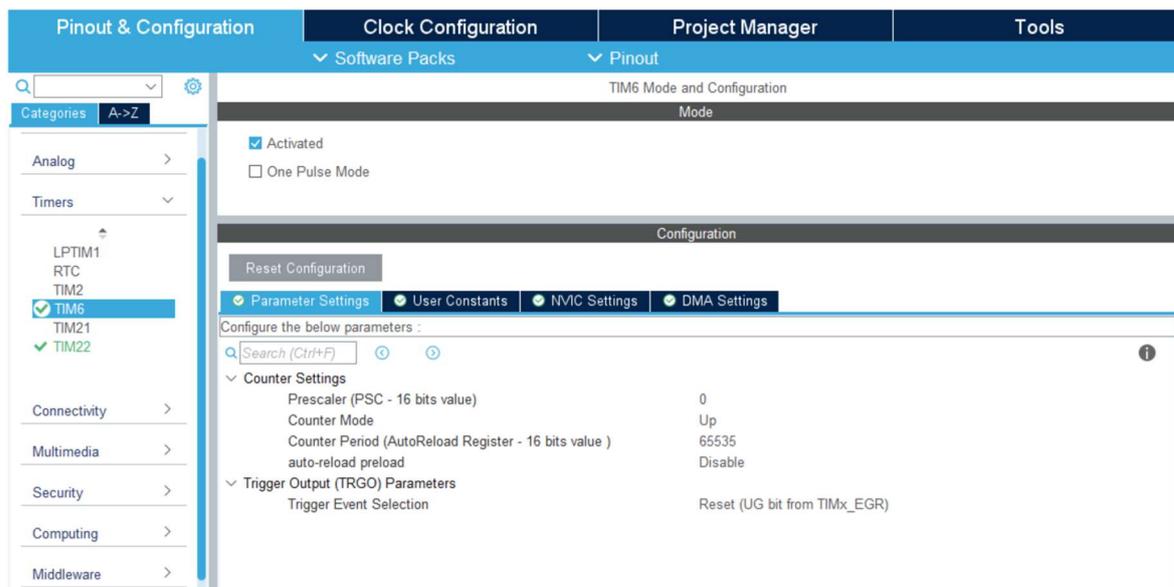


Figura 93. CubeMX Activación TIM6

Para el uso del *driver* se activarán las salidas digitales mencionadas (A10 y C7) como GPIO Output.

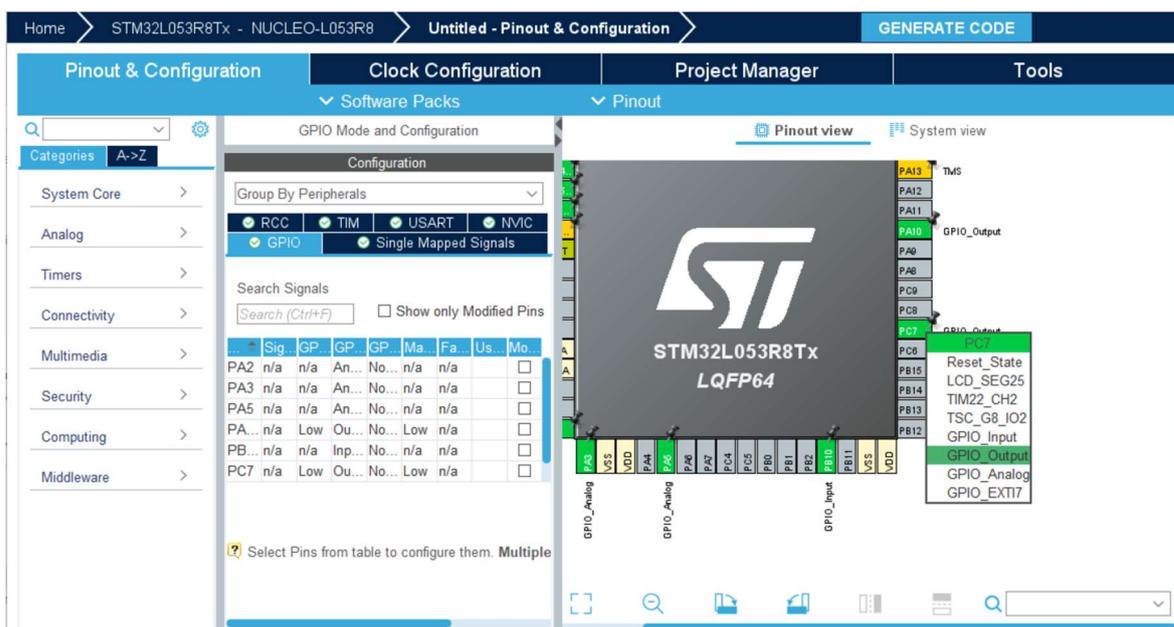


Figura 94. CubeMX selección pines salida digital control del driver

Por otro lado, la interrupción de la medida del *encoder* se asignará al pin C8, que se iniciará con la opción GPIO_EXTI_8. El botón azul también es de este tipo, pero ya viene predefinido en la configuración inicial por defecto.

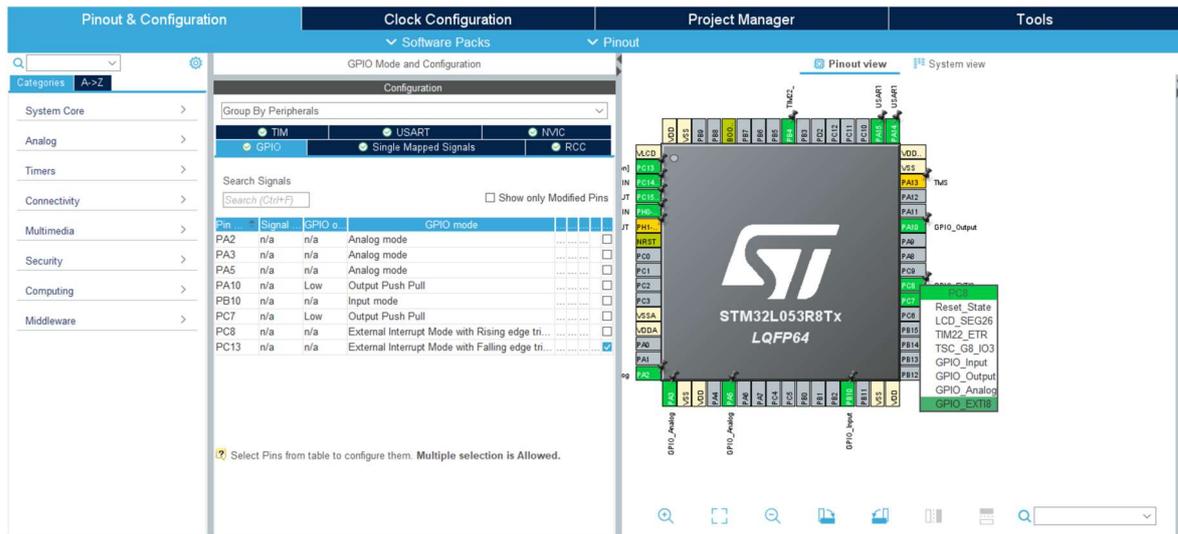


Figura 95. CubeMX interrupción pin C8

Con esto, a falta de lo que se programará más adelante, se tiene una configuración inicial con la que comenzar a trabajar.

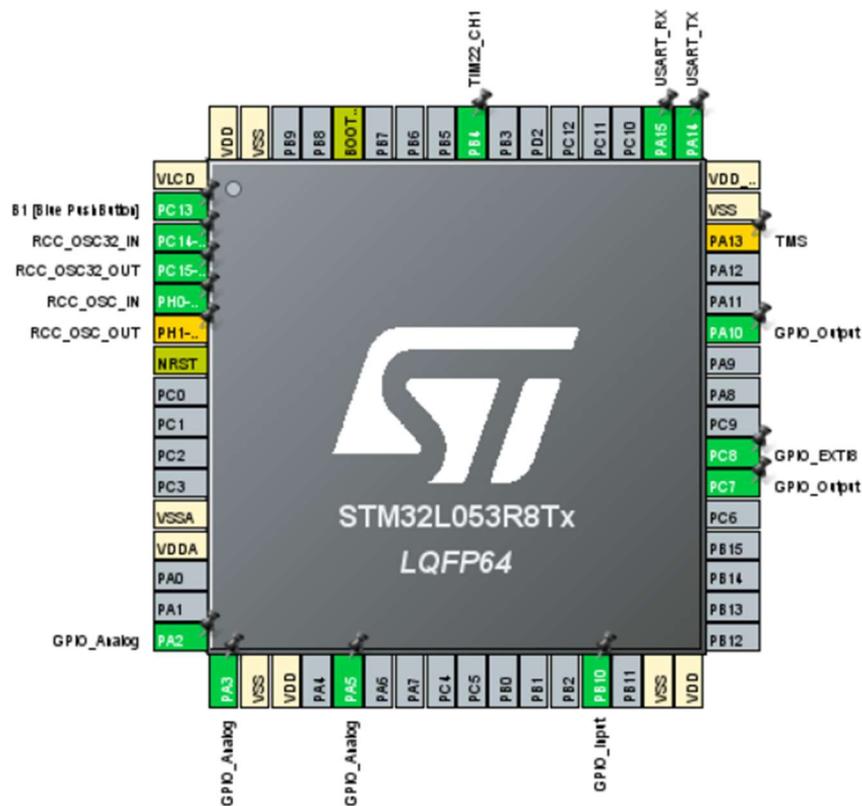


Figura 96. CubeMX configuración final pines del microcontrolador

Una vez terminada la configuración de pines, es posible configurar el reloj del microcontrolador desde el propio programa. Se ha seleccionado un reloj interno de 32 MHz.

En último lugar, se ha de generar el proyecto con el nombre y ubicación deseados desde la ventana “*Project Manager*”. Se puede realizar una exportación a varios IDEs disponibles; en este caso se va a trabajar en Keil μ Vision 5. Pulsando en el botón “GENERATE CODE” se genera un código con las librerías HAL necesarias para el funcionamiento y la composición e inicialización de pines realizada.

6.2.1.2 Keil μ Vision 5

Partiendo del proyecto base generado en la plataforma CubeMX, se va a programar el microcontrolador para que realice varias tareas: la lectura de las señales de entrada analógicas, su procesado, el envío por protocolo UART de comandos AT para configurar y transferirle los datos al módulo wifi y la generación de la señal PWM para controlar el motor que hace las veces de ventilador.

Para la programación, como ya se ha mencionado, se utilizará el IDE Keil μ Vision 5. Además, se aprovecharán algunas funciones incluidas en sus librerías y las de STM32Cube, HAL en particular, así como algunas desarrolladas por ST para la configuración de los módulos externos.

A continuación, se resumen algunas de las principales partes y funciones del código desarrollado.

6.2.1.2.1 Toma de valores de CO

Mediante la función *calculoCO()* se realiza la lectura de los pines A2, A3 y A5 y la aplicación de las fórmulas para el cálculo de concentración de gas CO y de la temperatura extraídas de las hojas de datos de los fabricantes de los sensores mencionadas anteriormente. Esta será llamada cada vez que se quiera medir el valor de monóxido de carbono.

```
float calculoCO()
{
    valorCO = Get_Gas_concentration((float)*pGain, ((float)*pSens)/1000);
    temperatura = Get_Temperature();
    valorCO_ppm_compensado =
    Get_comp_gas_concentration(valorCO, (int8_t)round(temperatura));

    return valorCO_ppm_compensado;
}
```

Como ya se ha comentado, ST ha desarrollado unas librerías especializadas para la lectura de entradas analógicas y la calibración del sensor a partir de lo indicado por el fabricante en su hoja de datos. Por tanto, se van a aprovechar estas funciones:

```
float Get_Gas_concentration(float gain, float sens)
{
    return ((float)Get_ADC_value(ANALOG_GAS_VALUE_PIN)
    Get_ADC_value(ANALOG_REF_VOLT_VALUE_PIN))*805664/gain/sens;
}

float Get_Temperature(void)
{
    return (-1481.96+sqrt(2196200+(1.8639
    3.3*Get_ADC_value(ANALOG_TEMP_VALUE)/4096)/3.88*1000000));
}

float Get_comp_gas_concentration(float gas_ppm, int8_t temp)
{
    //Se busca el coeficiente CF de compensacion de temperatura en una
    tabla (extraida del datasheet)
    if(temp+COMP_SHIFT>=LOOKUP_LENGTH)
    {
        temp=LOOKUP_LENGTH-1;
    }else if(temp+COMP_SHIFT<0)
    {
        temp=0;
    }else
    {
        temp=temp+COMP_SHIFT;
    }

    return gas_ppm/TEMP_COMPENSATION[temp];
}
```

La lectura de los pines analógicos se hace mediante las funciones `HAL_ADC_Start()` y `HAL_ADC_PollForConversion()` puesto que se va a emplear una técnica de conversión analógico-digital conocida como *polling*. Con ella, a diferencia de usar interrupciones, al iniciar la conversión ADC se para la CPU hasta que se completa la toma de muestras o conversión. Cuando termina, luego continua la ejecución del código principal. De este modo, es posible asegurarse de que se ha terminado la conversión y no se corrompen los datos o se toman valores anteriores.

6.2.1.2.2 Conexión wifi

Para conectar y configurar el módulo wifi para la transmisión de datos a la aplicación se utiliza la función `ESP_Init`.

Como el envío de información se hace por protocolo UART, se utiliza una función de la librería HAL, `HAL_UART_Transmit`, la cual requiere de cuatro parámetros: el identificador UART previamente definido, un puntero que apunta al *buffer* que almacena el mensaje que se quiere enviar, su tamaño y la duración del *timeout* (tiempo de espera hasta enviar).

```
HAL_UART_Transmit(wifi_uart, (uint8_t*)"AT+RST\r\n",
strlen("AT+RST\r\n"), HAL_MAX_DELAY);
```

La función *ESP_Init* es llamada con cuatro parámetros ("SSID", "contraseña", "IPv4" y "puerto") y se utiliza para iniciar el chip ESP8266. Esta envía varios comandos AT para reiniciarlo y desconectarlo de cualquier red anterior, asignar el modo de funcionamiento de estación, conectarlo a una red wifi cuyo nombre y contraseña se pasan como parámetros, conectarlo a continuación a un servidor cuya IP y puerto también se indican en el argumento y comenzar la transmisión de datos. Para hacer todo esto, por tanto, se envían 7 comandos AT seguidos: AT+RST, AT+CWQAP, AT+CWMODE_CUR, AT+CWJAP, AT+CIPSTART, AT+CIPMODE y AT+CIPSEND. La definición de cada uno de ellos está descrita en la sección de Comandos AT anterior.

```
void ESP_Init (char *SSID, char *PASSWD, char *IPv4_server, char
*PORT_server)
{
    char buffer[70];

    HAL_UART_Transmit(wifi_uart, (uint8_t*)"AT+RST\r\n",
strlen("AT+RST\r\n"), HAL_MAX_DELAY);
    HAL_Delay(1000);
    HAL_UART_Transmit(wifi_uart, (uint8_t*)"AT+CWQAP\r\n",
strlen("AT+CWQAP\r\n"), HAL_MAX_DELAY);
    HAL_Delay(1000);
    HAL_UART_Transmit(wifi_uart, (uint8_t*)"AT+CWMODE_CUR=1\r\n",
strlen("AT+CWMODE_CUR=1\r\n"), HAL_MAX_DELAY);
    HAL_Delay(1000);
    sprintf (buffer, "AT+CWJAP=\"%s\", \"%s\"\r\n", SSID, PASSWD);
    HAL_UART_Transmit(wifi_uart, (uint8_t*)buffer, strlen(buffer),
HAL_MAX_DELAY);
    HAL_Delay(6000);
    sprintf (buffer, "AT+CIPSTART=\"%TCP\", \"%s\", %s\r\n", IPv4_server,
PORT_server);
    HAL_UART_Transmit(wifi_uart, (uint8_t*)buffer, strlen(buffer),
HAL_MAX_DELAY);
    HAL_Delay(5000);
    HAL_UART_Transmit(wifi_uart, (uint8_t*)"AT+CIPMODE=1\r\n",
strlen("AT+CIPMODE=1\r\n"), HAL_MAX_DELAY);
    HAL_Delay(2000);
    HAL_UART_Transmit(wifi_uart, (uint8_t*)"AT+CIPSEND\r\n",
strlen("AT+CIPSEND\r\n"), HAL_MAX_DELAY);
    HAL_Delay(2000);
}
```

Puesto que algunos mensajes utilizan parámetros externos, primero se convierten en un *string* que se guarda en una variable *buffer* temporal que es la que luego se transmite.

Una vez se han enviado los comandos anteriores, los siguientes datos enviados por el pin UART el chip wifi los tratará como datos de texto, por lo que ya no se van a ejecutar más comandos AT.

6.2.1.2.3 Control del motor

Para utilizar la señal PWM lo primero es configurarla correctamente. Desde CubeMX se genera una función para ajustar el Timer seleccionado, el TIM22 en este caso, desde la cual se pueden asignar o cambiar sus parámetros.

Los valores de registro que hay que ajustar en el código serán los anteriormente calculados: 3999 de auto recarga y 0 de *prescaler*. Luego, para que el motor inicie el giro con un ciclo de trabajo del 100%, para el parámetro *pulse* se pondrá el valor de contador de registro máximo:

$$CCRx = (1 \cdot (3999 + 1)) - 1 = 3999 = ARR$$

```
htim22.Init.Prescaler = 0;
htim22.Init.CounterMode = TIM_COUNTERMODE_UP;
htim22.Init.Period = 3999; // ARR
sConfigOC.Pulse = 3999;
```

Asimismo, también se configuran los pines de controlador (*enable*, *standby* y dirección puente A) y se inicializan (a nivel alto, alto y bajo, respectivamente).

```
static void configuracion_modulo_motor(void)
{
    GPIO_InitTypeDef GPIO_InitStructure;

    //Stspin250 - pin ENABLE
    GPIO_InitStructure.Pin = CONTROL_MOTOR_EN_FAULT_PIN;
    GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStructure.Pull = GPIO_NOPULL;
    GPIO_InitStructure.Speed = GPIO_SPEED_MEDIUM;
    HAL_GPIO_Init(CONTROL_DIR_MOTOR_A_PUERTO, &GPIO_InitStructure);
    HAL_GPIO_WritePin(CONTROL_MOTOR_EN_FAULT_PUERTO,
        CONTROL_MOTOR_EN_FAULT_PIN, GPIO_PIN_SET);

    //Stspin250 - pin dirección puente A
    GPIO_InitStructure.Pin = CONTROL_DIR_MOTOR_A_PIN ;
    GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStructure.Pull = GPIO_NOPULL;
```

```
GPIO_InitStruct.Speed = GPIO_SPEED_MEDIUM;
HAL_GPIO_Init(CONTROL_DIR_MOTOR_A_PUERTO , &GPIO_InitStruct);
HAL_GPIO_WritePin(CONTROL_DIR_MOTOR_A_PUERTO, CONTROL_DIR_MOTOR_A_PIN,
GPIO_PIN_SET);

//Stspin250 - pin STAND BY/RESET
GPIO_InitStruct.Pin = CONTROL_MOTOR_RESET_PIN;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_MEDIUM;
HAL_GPIO_Init(CONTROL_MOTOR_RESET_PUERTO, &GPIO_InitStruct);
HAL_GPIO_WritePin(CONTROL_MOTOR_RESET_PUERTO, CONTROL_MOTOR_RESET_PIN,
GPIO_PIN_SET);
}
```

La señal PWM comenzará a emitirse en el pin correspondiente a nivel máximo a partir del momento en que se llama a la función HAL *HAL_TIM_PWM_Start*. Luego, cada vez que se quiera modificar la velocidad de giro del motor, se puede utilizar la expresión TIM22 -> CCR1, pasándole el valor de registro calculado según el ciclo de trabajo.

```
TIM22->CCR1 = 1499; // motor 75% del voltaje
```

Cuando se quiera detener el giro se puede poner el pin *standby* (PC7) a nivel bajo. Esta es una opción superior que simplemente bajar el ciclo de trabajo al 0%, puesto que el motor deja de estar alimentado y se reduce el consumo.

6.2.1.2.4 Interrupciones EXTI del sistema

Para que la CPU no tenga que estar constantemente leyendo el estado de algunos pines digitales, se utilizan las interrupciones, las cuales notifican a la CPU únicamente cuando sucede un evento determinado, de modo que esta detiene la ejecución del paso en el que se encuentre para pasar a ejecutar un código para tratarlo. Para gestionarlas, los microcontroladores STM32 utilizan el periférico EXTI.

En este caso, se han programado dos interrupciones. En cuanto a la primera, está vinculada al botón azul USER incorporado en la placa. Al pulsarlo, cambiará el valor del pin *reset* del driver, modificando el estado del motor (cuando el pin está en estado bajo, el driver deja de alimentar al motor y viceversa).

La segunda está relacionada con el *encoder*. Cada vez que este completa una vuelta, la señal índice (X en el modelo utilizado) genera un pulso de nivel alto. Cuando esto suceda, saltará la interrupción y aumentará en 1 un contador de vueltas, que luego se utilizará para determinar la velocidad a la que está girando el motor.

```
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)//Interrupciones
{
    if (GPIO_Pin == B1_Pin) //botón azul
    {
        HAL_GPIO_TogglePin(CONTROL_MOTOR_RESET_PUERTO,
            CONTROL_MOTOR_RESET_PIN);
    }
    if (GPIO_Pin == Index_encoder_Pin) //índice encoder
    {
        contador_vueltas++;
    }
}
```

La configuración de estos pines como interrupción se hace en la función `MX_GPIO_init`, creada automáticamente a partir de lo hecho en CubeMX del siguiente modo:

```
//Inicializar el boton de entrada azul como interrupcion
HAL_NVIC_DisableIRQ(EXTI0_1_IRQn);
GPIO_InitTypeDef pin1;
pin1.Pin = B1_Pin;
pin1.Mode = GPIO_MODE_IT_FALLING;
pin1.Pull = GPIO_NOPULL;
HAL_GPIO_Init(B1_GPIO_Port, &pin1);
HAL_NVIC_SetPriority(EXTI0_1_IRQn,0,0);
HAL_NVIC_EnableIRQ(EXTI0_1_IRQn);

//Inicializar el pin C8 como interrupcion
HAL_NVIC_DisableIRQ(EXTI4_15_IRQn);
GPIO_InitTypeDef pin2;
pin2.Pin = Index_encoder_Pin;
pin2.Mode = GPIO_MODE_IT_FALLING;
pin2.Pull = GPIO_NOPULL;
HAL_GPIO_Init(Index_encoder_Port, &pin2);
HAL_NVIC_SetPriority(EXTI4_15_IRQn,0,2);
HAL_NVIC_EnableIRQ(EXTI4_15_IRQn);
```

6.2.1.2.5 Tarea periódica: medición, envío y activación del motor

Puesto que se busca medir la concentración de monóxido de carbono y enviar por wifi el valor detectado por el sensor cada cierto tiempo, se va a programar una tarea periódica. Una manera precisa de hacerlo es mediante una interrupción generada por uno de los *timers* del microcontrolador, el TIM6.

TIM 6 es un temporizador de 16 bits de cuenta ascendente básico. Se puede utilizar para realizar conversiones digital-analógicas, pero en este caso se utilizará de contador. Cuando este llegue a su límite, se va a programar una interrupción que llamará a la función del proceso de toma de medidas y transmisión UART.

Cuando la interrupción se activa, el sistema llama a la función *HAL_TIM_IRQHandler*, pasándole como parámetro el nombre del temporizador en cuestión. Desde aquí se llama a *HAL_TIM_PeriodElapsedCallback*, que contiene el código que se ejecutará cada vez que el contador llegue a su fin, en este caso la tarea periódica. Por tanto, cada vez que pase un tiempo asignado en el *timer*, 15 segundos, para poder probar el programa con comodidad y ver cambiar los valores, se llamará a las funciones de lectura de sensores y de transmisión por protocolo UART.

```
//Inicializar la interrupcion del TIMER6
HAL_NVIC_DisableIRQ(TIM6_IRQn);
HAL_NVIC_SetPriority(TIM6_IRQn, 0, 1);
HAL_NVIC_EnableIRQ(TIM6_IRQn);
```

```
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef* htim)
{
    if(htim == &htim6)
    {
        proceso_Completo();
    }
}
```

Para adaptar el tiempo que tarda el *timer* en completar su ciclo de cuenta, se han de ajustar su valor de auto registro y su *prescaler*. El primero establece el valor hasta el cual el temporizador contará antes de reiniciar mientras que el *prescaler* sirve para determinar la frecuencia del contador del temporizador. Para calcular la frecuencia de interrupción, se puede utilizar la misma fórmula que se ha visto en el apartado anterior (6.1.3.1 Controlador y motor):

$$f_{\text{interrupción}} = \frac{f_{TIM}}{(PSC + 1) \cdot (ARR + 1)}$$

$$T_{interrupción} = \frac{(PSC + 1) \cdot (ARR + 1)}{f_{TIM}}$$

Sabiendo que la frecuencia del reloj es la interna del sistema (*SystemCoreClock*, 32 MHz) y que se busca una interrupción cada 15 segundos, se puede escoger un valor de *prescaler* dentro de su rango (1 a 65536 [23]) y despejar así el valor de auto registro:

$$ARR = \frac{f_{TIM} \cdot T_{interrupción}}{(PSC + 1)} - 1 = \frac{32 \cdot 10^6 \cdot 15}{(9999 + 1)} - 1 = 47999$$

Puesto que 47999 es un número de 16 bits, se podrá utilizar [37].

```
htim6.Init.Prescaler = 9999;  
htim6.Init.CounterMode = TIM_COUNTERMODE_UP;  
htim6.Init.Period = SystemCoreClock*15/(9999+1)-1;
```

La función `proceso_Completo()`, que se ejecuta cada vez que el temporizador alcanza su valor final, se puede dividir en dos partes: el envío de la medida de CO por protocolo UART y la activación del ventilador.

Por un lado, como se ha mencionado, se desea realizar la medición y el envío de los datos de forma periódica, utilizando la interrupción del temporizador 6 para ello. En el momento que el contador llegue a su valor final, se ejecutarán una serie de instrucciones para leer los valores de los sensores, calcular la concentración de CO en ppm y enviar este dato por el pin de transmisión UART.

Por otro lado, habrá una segunda secuencia condicional en la que, si el valor de monóxido de carbono mencionado supera un valor límite establecido, se activará el *driver* para controlar el motor, poniéndose este a girar a cierta velocidad dependiendo de cómo de alta sea la concentración de dicho gas (a mayor CO, mayor velocidad). En la Figura 97 se puede ver un esquema que representa este funcionamiento, estando definidos los límites de CO mencionados en ella, establecidos según los niveles saludables para el ser humano, al inicio del código.

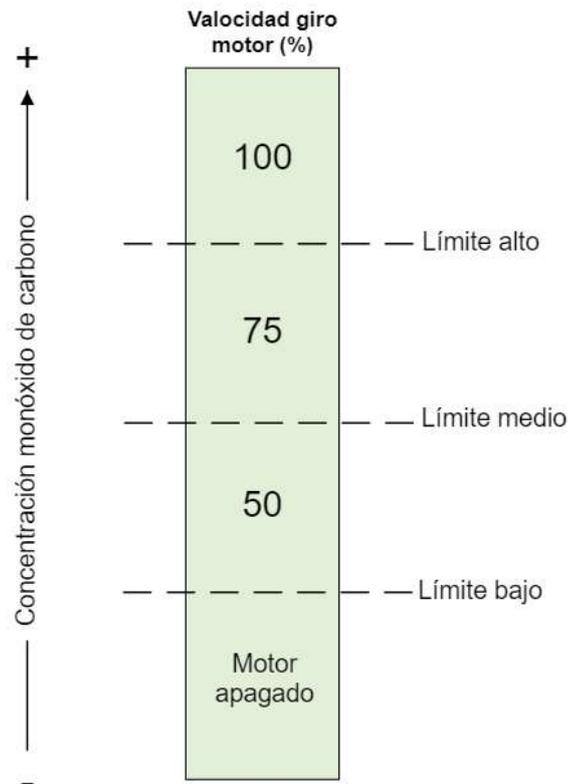


Figura 97. Esquema de la velocidad de giro del motor según la cantidad de CO medido. Elaboración propia.

```
void proceso_Completo(void)
{
    char buf[10];
    valorCOppm = calculoCO(); //lectura de CO
    sprintf (buf, "%.2f\r\n", valorCOppm); //convertir entero a string
    HAL_UART_Transmit(wifi_uart, (uint8_t*)buf, strlen(buf),
    HAL_MAX_DELAY);

    //si la concentración de CO supera el limite seguro -> encender el
    Ventilador
    if (valorCOppm > limiteCO_bajo)
    {
        //activar driver
        HAL_GPIO_WritePin(CONTROL_MOTOR_RESET_PUERTO,CONTROL_MOTOR_RESET_
        PIN, GPIO_PIN_SET);

        if (limiteCO_bajo <= valorCOppm & valorCOppm < limiteCO_medio){
            // motor 50% del voltaje
            TIM22->CCR1 = 1999;
        }
        else if(limiteCO_medio <= valorCOppm & valorCOppm<limiteCO_alto){
            // motor 75% del voltaje
            TIM22->CCR1 = 2999;
        }
        else if (limiteCO_alto <= valorCOppm){
            // motor 100% del voltaje
```

```
TIM22->CCR1 = 3999;
}

//si el valor de CO es bajo, apagar el ventilador
} else HAL_GPIO_WritePin(CONTROL_MOTOR_RESET_PUERTO,
CONTROL_MOTOR_RESET_PIN, GPIO_PIN_RESET);

mostrar_velocidad_motor();
}
```

En último lugar, puesto que se busca determinar la velocidad de giro del motor en revoluciones por minuto, se lee el número de veces que ha girado el codificador rotatorio en el último periodo, se multiplica por 60 y se divide entre el tiempo que dura cada uno de los periodos de lectura. Luego, se envía este valor por el puerto serie y se reinicia el contador del *encoder*.

```
void mostrar_velocidad_motor(void)
{
    uint32_t velocidad_motor;
    char buf[10];

    //convertir las vueltas dadas en un periodo a rpm
    velocidad_motor = contador_vueltas*60/periodo_medidas;
    //Convertir a string con nna M delante para que la aplicación
    distinga entre velocidad y CO
    sprintf (buf, "M%d\r\n", velocidad_motor);
    //Enviar por UART
    HAL_UART_Transmit(wifi_uart, (uint8_t*)buf, strlen(buf),
    HAL_MAX_DELAY);

    //reiniciar contador
    contador_vueltas = 0;
}
```

El mensaje se envía con una letra M delante de la medida para que luego, como todas llegan al servidor del mismo modo, la aplicación distinga entre las medidas de CO y las de velocidad.

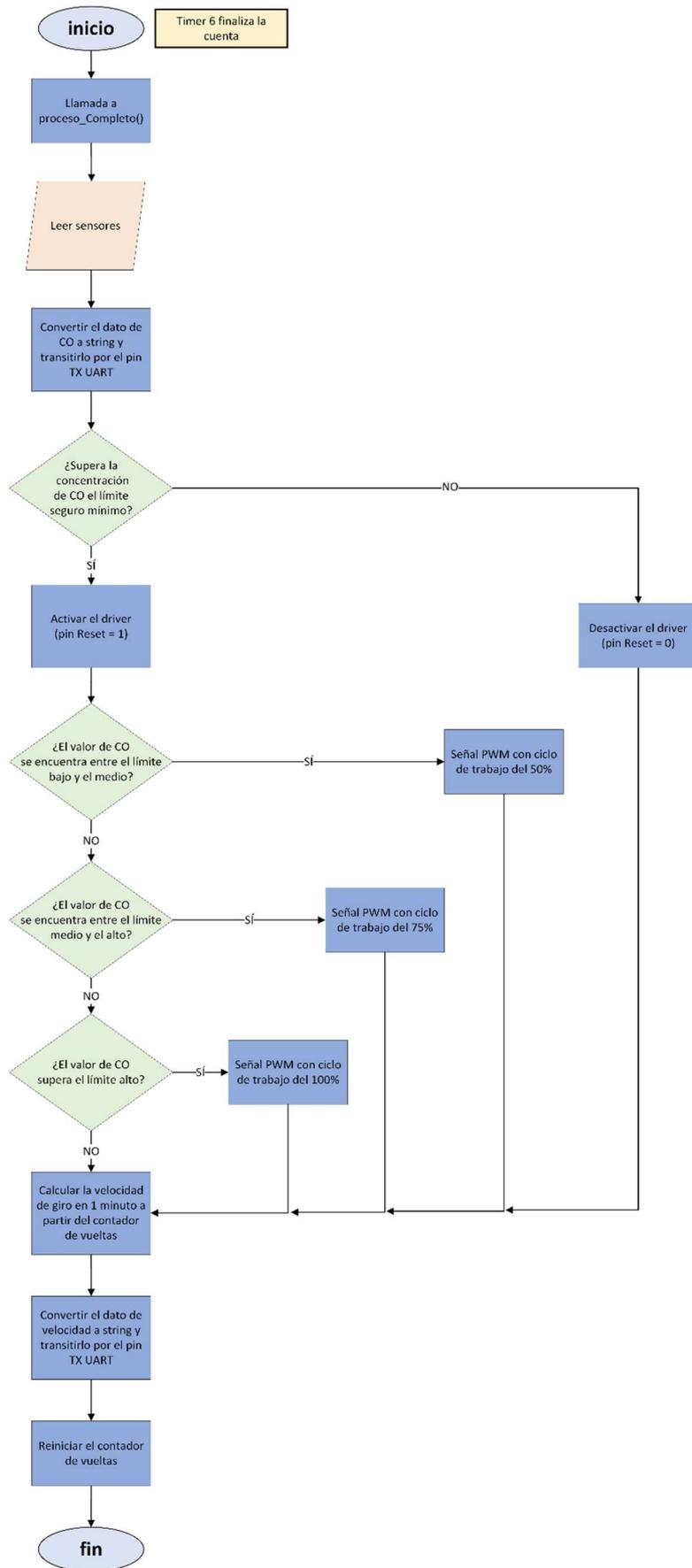


Figura 98. Diagrama de flujo del funcionamiento de la función proceso_Completo()

6.2.1.2.6 Función principal. Main().

El sistema llama a la función principal main() cuando el programa comienza su ejecución.

En primer lugar, se llama a tres funciones predefinidas de la librería HAL para configurar el microcontrolador.

- Hal_init: Configura la flash y el buffer cada vez que se reinicia el programa.
- SystemClock_Config: función predefinida. Configura el reloj interno del MCU e inicializa el oscilador.
- MX_GPIO_Init: función predefinida. configuración de los pines GPIO seleccionados.

Junto a ellas también se llama a las funciones para configurar el *timer* para generar la señal PWM (MX_TIM22_Init) y los pines UART (MX_USART2_UART_Init) según las especificaciones seleccionadas.

A continuación, se configuran los pines para el manejo del driver de control de motores mediante la función configuracion_modulo_motor() y se inicia la generación de la señal PWM, la cual se dejará 3 segundos haciendo girar el motor a máxima velocidad antes de apagarlo poniendo el pin *reset* del controlador a 0 para comprobar que rota correctamente.

Seguidamente, se ejecutará la función ya mencionada de configuración del chip wifi y, para que comience el contador que activa la tarea de toma y envío de medidas, se llama a una serie de funciones para configurar el timer 6, iniciarlo y accionar sus interrupciones.

Tras todas las inicializaciones, se llama a la función "*while (1)*", que se repite cíclicamente mientras el microcontrolador esté encendido. Sin embargo, esta función no contendrá código, el proceso se realizará exclusivamente con interrupciones del temporizador, de modo que se libera espacio de la CPU.

```
int main(void)
{
    HAL_Init();
    SystemClock_Config();

    MX_GPIO_Init();
    MX_TIM22_Init();
    MX_USART2_UART_Init();

    configuracion_modulo_motor();
    //Iniciar el giro del motor a máxima velocidad y luego pararlo
    HAL_TIM_PWM_Start(&htim22,TIM_CHANNEL_1);
    HAL_Delay(3000);
    HAL_GPIO_WritePin(CONTROL_MOTOR_RESET_PUERTO, CONTROL_MOTOR_RESET_PIN,
    GPIO_PIN_RESET);
```

```
//Comandos AT módulo wifi
ESP_Init("PuntoWifi","12345678","192.168.1.46","1024");

//Inciar el timer que gestiona el proceso completo
MX_TIM6_Init();
HAL_TIM_Base_MspInit(&htim6);
HAL_TIM_Base_Init(&htim6);
HAL_TIM_Base_Start_IT(&htim6);

while (1)
{
}
}
```

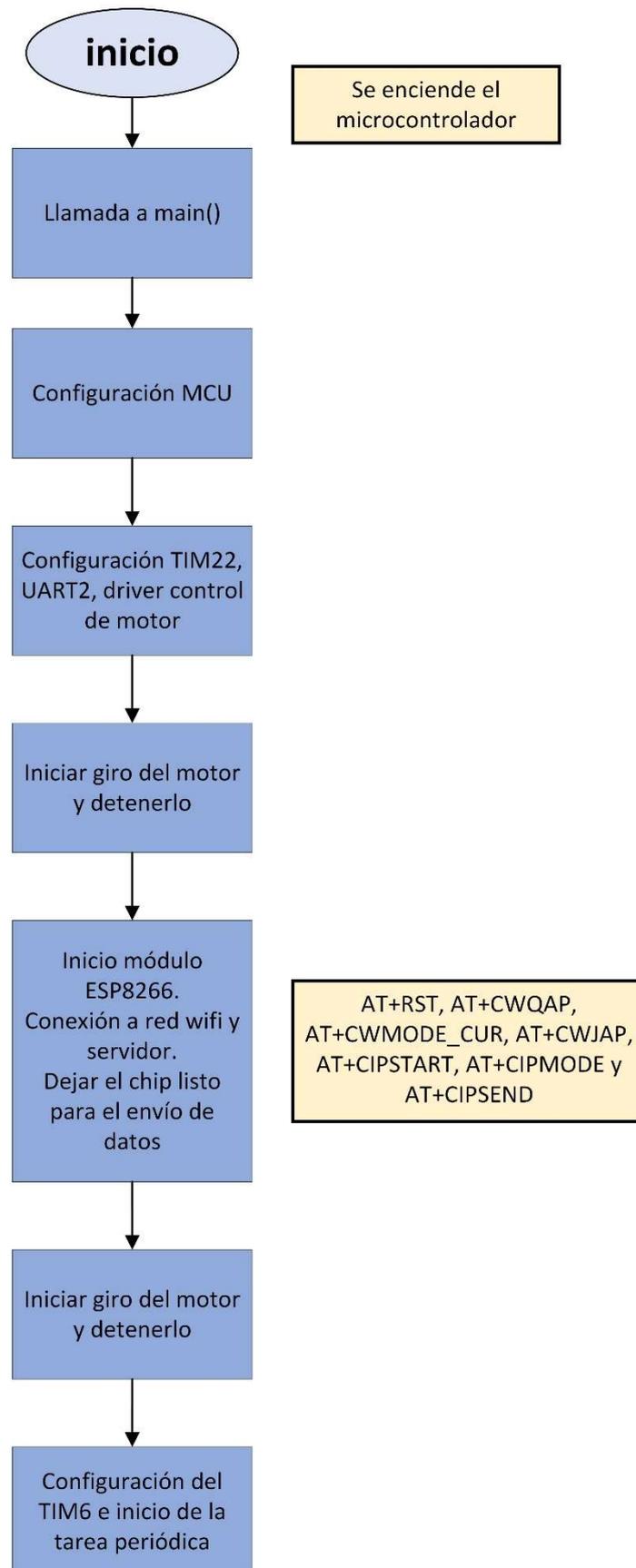


Figura 99. Diagrama de flujo del funcionamiento de la función main()

6.2.2 Visualización de la información. Servidor e interfaz gráfica

Para ver las mediciones de monóxido de carbono, el módulo wifi se conectará como cliente a un servidor abierto por el usuario en una aplicación externa y enviará los datos en un periodo de tiempo predefinido.

El funcionamiento del programa en QT se resume en el diagrama de la Figura 100.

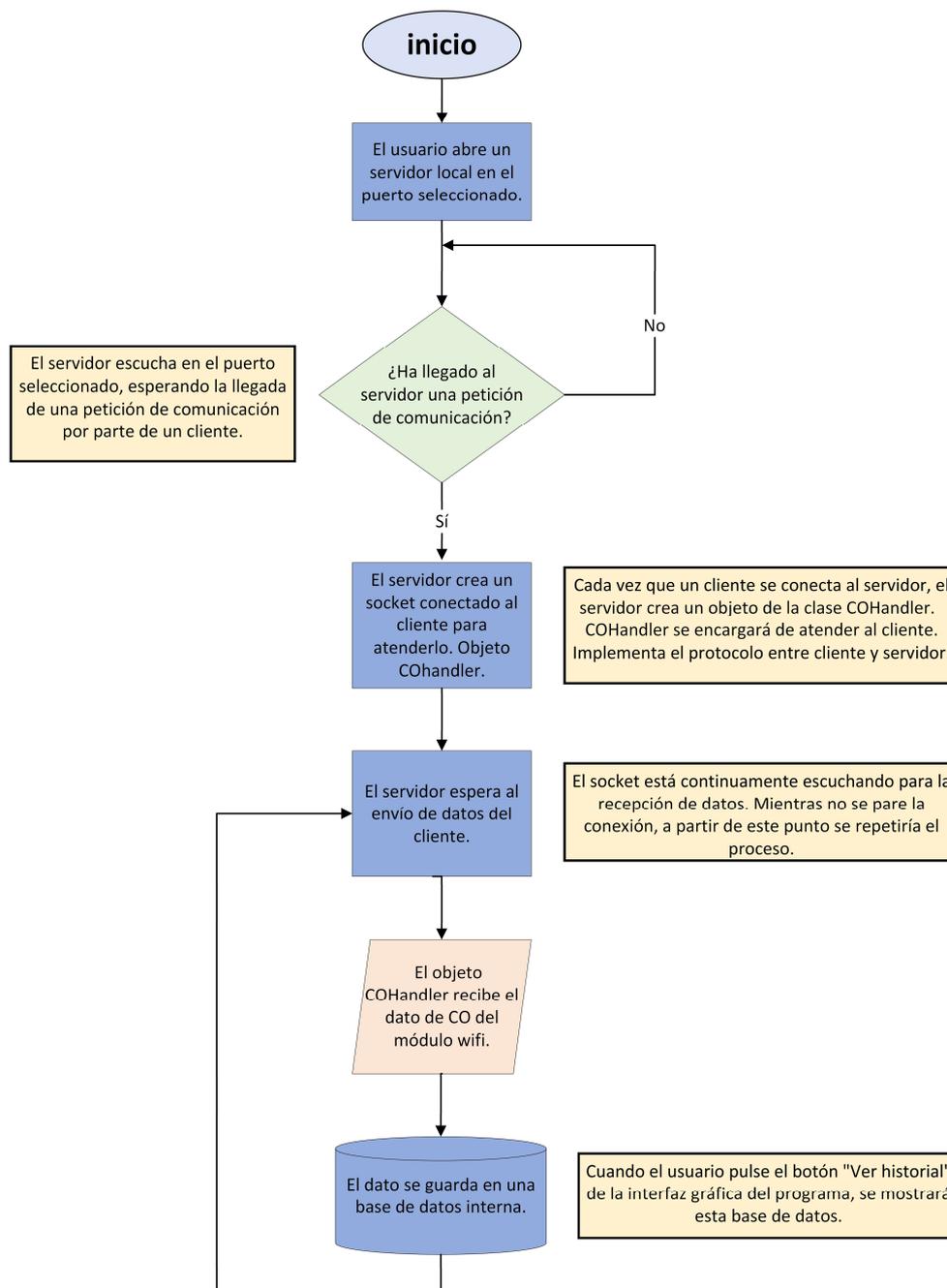


Figura 100. Diagrama de flujo del funcionamiento de la aplicación diseñada en QT para recibir datos del dispositivo

Básicamente, el proceso comienza cuando el usuario, desde la pantalla principal de la interfaz gráfica, pulsa un botón para abrir el servidor. Este está configurado para escuchar en la red en la que se encuentra el ordenador y en un puerto específico programado de antemano, que coincide con el que se le escribe al chip wifi por comando desde el microcontrolador.

El servidor es un objeto de la clase QTcpServer de QT. Para ponerlo en marcha, se utiliza el método *listen()* con 2 parámetros: la red y el puerto TCP. El servidor estará en continuo funcionamiento esperando peticiones de conexión.

```
CServer::CServer(quint16 p, QObject *parent): QObject{parent}
{
    //Hacer que el valor del puerto sea privado para que no se pueda
    modificar desde el programa.
    setPuerto(p);

    //crear el servidor nuevo
    _servidor = new QTcpServer();//Servidor tipo TCP

    //Poner el servidor a escuchar a través del puerto asignado. Se va a
    usar IPv4
    if (_servidor->listen(QHostAddress::AnyIPv4,GetPuerto()))
    {
        //Si el servidor ha podido abrir el puerto, conectar la señal con
        el slot
        connect(_servidor,SIGNAL(newConnection()),this,SLOT(conexionNueva()));
    }
    else
    {
        //Si el servidor no ha podido abrir el puerto, se destruye
        delete _servidor;
    }
}
```

Cuando un cliente finalmente establezca una conexión con el servidor, este creará un socket para atenderlo. Los sockets son canales de comunicación que permiten que dos procesos que no están relacionados intercambien datos, en este caso por la red. Para el manejo de sockets y el envío y recepción de datos a través de una conexión TCP se utiliza la clase QTcpSocket.

```
void CServer::conexionNueva()
{
    //slot que acepta nuevas conexiones de clientes y crea objetos de
    tipo CHandler por conexión
    while(_servidor->hasPendingConnections())
    {
        QTcpSocket *socket = _servidor->nextPendingConnection();
        CHandler *conexion = new CHandler(socket); //objeto tipo
        CHandler
    }
}
```

```

        //Conectar la señal de recepción de datos desde el handler con el
        //slot de recepción del servidor
        connect(conexion, &COhandler::recibirDatoDeESP8266,
        this,&COserver::recibirDatoDeHandler);

        //enviar la IP a mainWindow para poder mostrarla
        emit enviarIPaMain(conexion>direccionIpRemota().toString());

        //para detectar cuando se desconecta cada cliente:
        connect(conexion, SIGNAL(desconectado()), this,
        SLOT(clienteDesconectado()));
    }
}

```

Al aceptarse una conexión se crea un diálogo entre servidor y cliente. Para atender a cada cliente, se creará un objeto de la clase COHandler. Esta clase implementa el protocolo entre cliente y servidor, es decir, define los mensajes que van a intercambiar.

```

COhandler::COhandler(QTcpSocket *socket, QObject *parent) :
QObject{parent}
{
    _socket = socket;

    if (canalAbierto())
    {
        //conectar la señal al slot para leer datos cuando lleguen al
        //socket
        connect(socket, SIGNAL(readyRead()), this, SLOT(datosCliente()));

        //conectar las señales para emitir la señal "desconectado()"
        //cuando se cierre el socket
        connect(socket, SIGNAL(disconnected()), this, SIGNAL(desconectado()));

        //enviar primer mensaje del protocolo
        QString inicio = "Servidor medidas CO:\r\n";
        _socket->write(inicio.toUtf8());
        _socket->flush();
    }
}

```

Dentro de esta clase se conectará la señal de recepción de datos del socket con el slot DatosCliente(), de modo que este se ejecutará cada vez que se reciba algo por el puerto. Mientras no se pare la conexión, el socket estará continuamente escuchando. Además, cuando un cliente se conecte, recibirá el mensaje de bienvenida “Servidor medidas CO”.

Las medidas de velocidad del motor y de monóxido de carbono se han de mostrar al usuario por pantalla. Para ello, se empleará la metodología de “*Signals & Slots*” (señales y

ranuras) de QT: cada vez que llega una medida, el objeto COHandler emite una señal que recibe el servidor con el dato y este a su vez vuelve a emitirla para que la reciba MainWindow, que es el objeto que proporciona una ventana principal de la aplicación. Esto se hace debido a que MainWindow y COHandler no tienen comunicación directa entre ellos. En los siguientes extractos de código, así como la Figura 101, se puede ver el funcionamiento básico:

- En COhandler:

```
void COhandler::datosCliente()
{
    while (_socket->canReadLine())
    {
        //leer una linea del socket
        QByteArray buffer = _socket->readLine();
        QString linea = QString::fromUtf8(buffer);

        //Se emite la señal con el valor medido vinculada a un Slot del
        servidor para que lo reciba.
        emit recibirDatoDeESP8266(linea);

        //Para ver las medidas por el terminal:
        QString reply;

        if(linea[0]== 'M')
        {
            //si el dato recibido empieza por M, se trata del valor de
            velocidad a la que gira el motor
            for (int i = 0; i<linea.size();i++)
            {
                //Para no mostrar el primer carácter (la M):
                linea[i] = linea[i+1];
            }
            reply = "Velocidad (rpm): " + linea + "\r\n";
        }
        else
        {
            //si el dato es el monóxido de carbono
            reply = "CO (ppm): " + linea + "\r\n";
        }

        //enviar la respuesta al terminal
        _socket->write(reply.toUtf8());
        _socket->flush();
    }
}
```

- En COserver:

```
void COserver::recibirDatoDeHandler(QString dato)
{
```

```
emit(COserver::enviarDatoDeHandlerAMain(dato);  
}
```

- Y, por último, en MainWindow:

```
void MainWindow::recibirDatoDeServer(QString dato)  
{  
    if(dato[0]== 'M')  
    {  
        //si el dato recibido empieza por M, se trata del valor de  
        //velocidad a la que gira el motor  
        for (int i = 0; i<dato.size();i++)  
        {  
            dato[i] = dato[i+1]; //No mostrar el primer caracter (la M)  
        }  
        //Se escribe la velocidad en su cuadro de texto correspondiente:  
        ui->lineEdit_rpm->setText(dato.trimmed() + "rpm");  
    }  
  
    else  
    {  
        //Añadir el nuevo valor de CO al historial de medidas  
        baseDeDatosCO.insertarValorEnHistorial(dato.toDouble());  
        ui->lineEdit_CO->setText(dato + "ppm");  
  
        if (dato.toDouble(>10) //Alarma en caso de CO alto  
        {  
            //Recuadro de la medida de CO en rojo (alarma)  
            ui->lineEdit_CO->setStyleSheet("background-color: red");  
        }  
        //Recuadro de la medida de CO en blanco (no alarma)  
        else {ui->lineEdit_CO->setStyleSheet("background-color: white");}  
  
        //Emitir el valor de CO como señal que recibirá el objeto de la  
        //gráfica  
        emit enviarDatoAGrafica(dato.toFloat());  
    }  
}
```

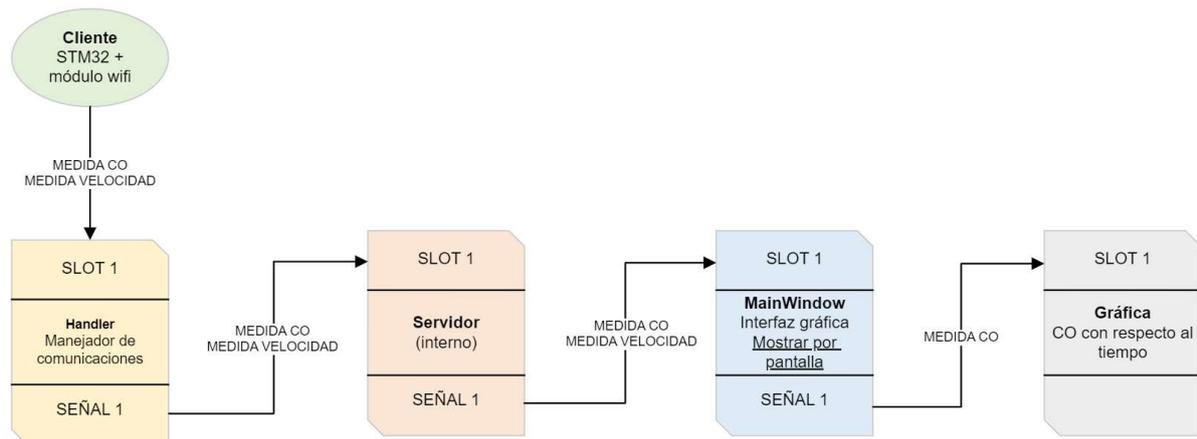


Figura 101. Diagrama de transmisión de datos dentro de las clases utilizadas en la aplicación de visualización

Una vez recibida la medida de monóxido de carbono, esta se almacenará, junto a su marca de tiempo, en una base de datos creada en la página principal (MainWindow) de modo que pueda ser consultada por el usuario cuando lo desee pulsando un botón. Para hacer esto se utilizan métodos de la clase QSqlQuery para componer y ejecutar sentencias SQL (el lenguaje para trabajar con bases de datos) sobre el sistema de gestión de bases de datos.

```

Void BaseDeDatosCO::insertarValorEnHistorial(double valor)
{
    QSqlQuery query(base_datos);

    query.prepare("INSERT INTO historial_medidas_CO (tiempo,valor)
VALUES (?,?)");
    query.addBindValue(QDateTime::currentDateTime());
    query.addBindValue(valor);
}

```

La base de datos creada en este caso es un objeto de la clase QSqlDatabase. Esta clase se utiliza principalmente para la conexión, configuración y establecimiento de una sesión con un sistema de gestión de bases de datos, en este caso de SQLite.

```

bool BaseDeDatosCO::conectar()
{
    base_datos = QSqlDatabase::addDatabase("SQLITE");
    base_datos.setDatabaseName("db_monoxido_carbono");
    return base_datos.open();
}

```

Por otro lado, se ha desarrollado una clase propia para generar un gráfico lineal, heredera de las clases de QT QWidget, QChart y QChartview. Esta se ha conectado con la señal de recepción de datos, de modo que se puede comprobar visualmente cómo varía el CO con el tiempo.

```
void Grafica::NuevaMuestra()
{
    static long muestra = 0;
    //con el static delante, se crea al invocarla, se inicializa a 0 y
    luego no desaparece

    serie->append(muestra, getDatos_CO());
    muestra++;
}
```

Las variables y funciones utilizadas para desarrollar el programa se almacenan en diferentes bibliotecas. Cada una contiene un conjunto de datos y las operaciones para manipularlos. El distribuir el programa en bibliotecas es útil tanto para mantener un mayor orden como para proteger ciertas variables y funciones y que no sean accesibles desde cualquier parte del programa.

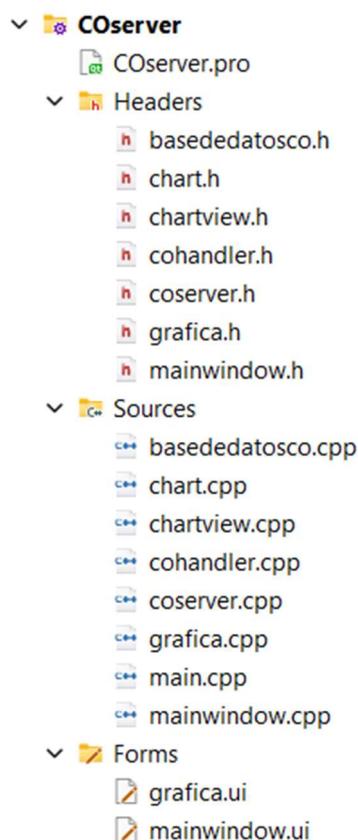


Figura 102. Bibliotecas usadas en la aplicación COserver

En cuanto la interfaz con la que se encontrará el usuario, esta consta de dos pestañas: la relacionada con el servidor y la relacionada con las medidas.

Respecto a la primera, Servidor, está compuesta de dos botones, uno para iniciar y otro para cerrar el servidor, una casilla con el puerto en el que comunicar y un cuadro de texto para recibir mensajes informativos en cuanto se abra el servidor, se cierre y se conecte algún cliente.



Figura 103. Interfaz gráfica: primera pestaña, Servidor

La segunda pestaña se denomina Medidas y es donde el usuario puede encontrar las lecturas de monóxido de carbono del sensor y de la velocidad del motor. En la parte superior hay dos cuadros de texto: el de la izquierda muestra la medición de CO más reciente y el de la derecha la velocidad de giro del motor en rpm. Más abajo, en un bloque de texto más grande, se puede encontrar el historial ordenado de valores de gas. Además, hay cuatro botones: el superior hace aparecer la información del historial, el segundo lo guarda en un archivo de texto externo (.txt), el tercero lo borra y el cuarto abre una ventana que contiene una gráfica que representa la variación de CO, en ppm, con respecto al tiempo.

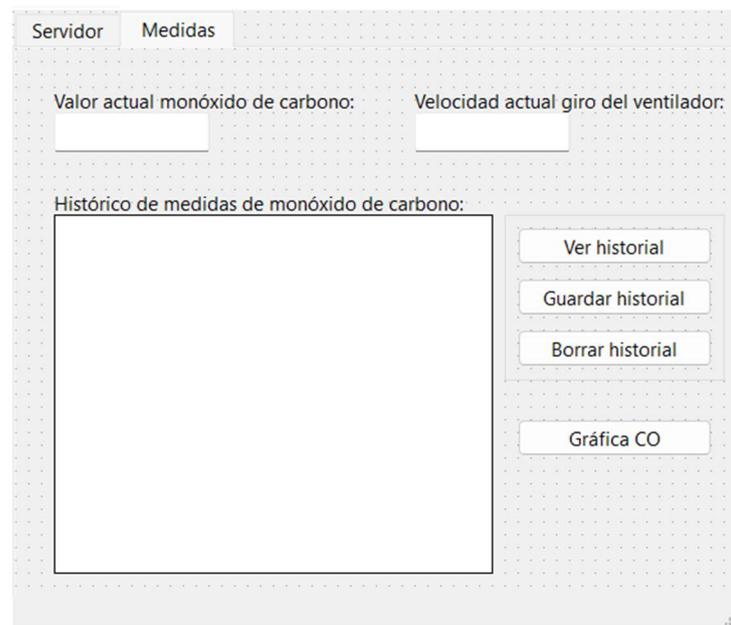


Figura 104. Interfaz gráfica: segunda pestaña, Medidas

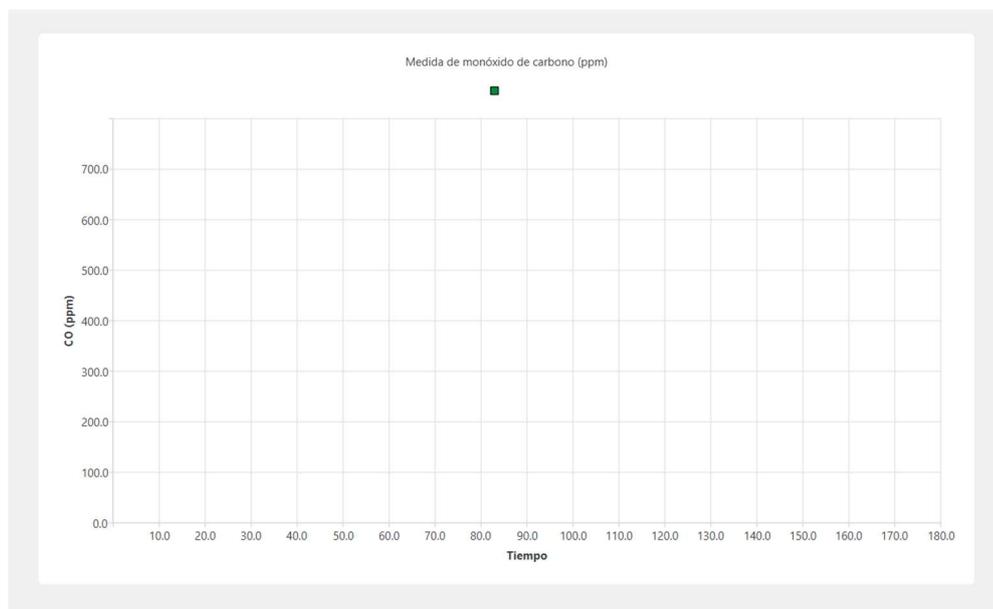


Figura 105. Interfaz gráfica: ventana gráfico de línea

Para programar el guardado de datos en un documento de texto, se crea un objeto de la clase QFile, en el que se escriben uno a uno los valores almacenados en la base de datos.

```
void MainWindow::on_pushButton_guardar_basededatos_clicked()
{
    QFile ficheroTexto("C:\\Universidad\\4\\TFG\\QT
    Servidor\\MedidasCO.txt");
    ficheroTexto.open(QIODevice::WriteOnly | QIODevice::Text);

    if (!baseDeDatosCOConectada) return;
}
```

```
QList<TuplaTiempoValor> *lista = baseDeDatosCO.leerHistorial();

ui->listWidget_medidas->clear();

for (int i = 0; i < lista->length(); i++)
{
    TuplaTiempoValor tupla = (*lista)[i];

    QString string = tupla.tiempo.toString("yyyy-MM-dd hh:mm:ss") +
        "-> CO: " + QString::number(tupla.valor) + " ppm";
    QTextStream escribirFichero(&ficheroTexto);
    escribirFichero << string + "\n";
}

delete lista;
}
```

7 PRUEBA Y ANÁLISIS DE RESULTADOS

Una vez completado el montaje y desarrollada la programación, tanto del microcontrolador como de la aplicación para ver la interfaz, se realiza una prueba para analizar el funcionamiento.

En cuanto a las condiciones externas, esta se hará en una habitación sin fuentes de monóxido de carbono cercanas, de modo que la concentración debería ser muy baja (generalmente menor a las 0,5 ppm), y a temperatura ambiente.

El microcontrolador y el resto de los componentes estarán alimentados desde la placa de prototipado, que tendrá en uno de sus extremos el módulo que transforma la entrada de 9 V de la fuente en dos salidas, una de 3,3 V y otra de 5. El adaptador convertidor AC/DC de 9 V estará conectado a una toma de corriente estándar.

En primer lugar, se ha abierto un terminal en Tera Term para poder leer el tiempo real los mensajes que envía y recibe el micro. Los parámetros de este terminal son los del apartado 6.1.2.2 Pruebas previas transmisión.

A continuación, desde la aplicación QT se ejecuta el programa desarrollado, apareciendo entonces la pantalla inicial de la aplicación que se puede ver en La Figura 106. Se pulsa el botón 'Abrir' para activar el servidor e iniciar la espera de peticiones de conexión de clientes en el puerto 1024.

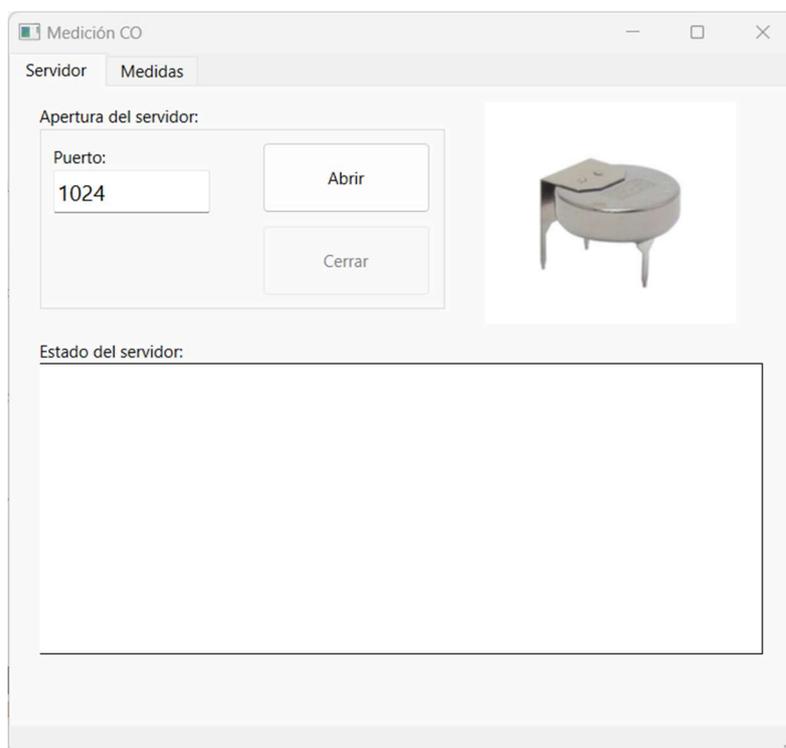
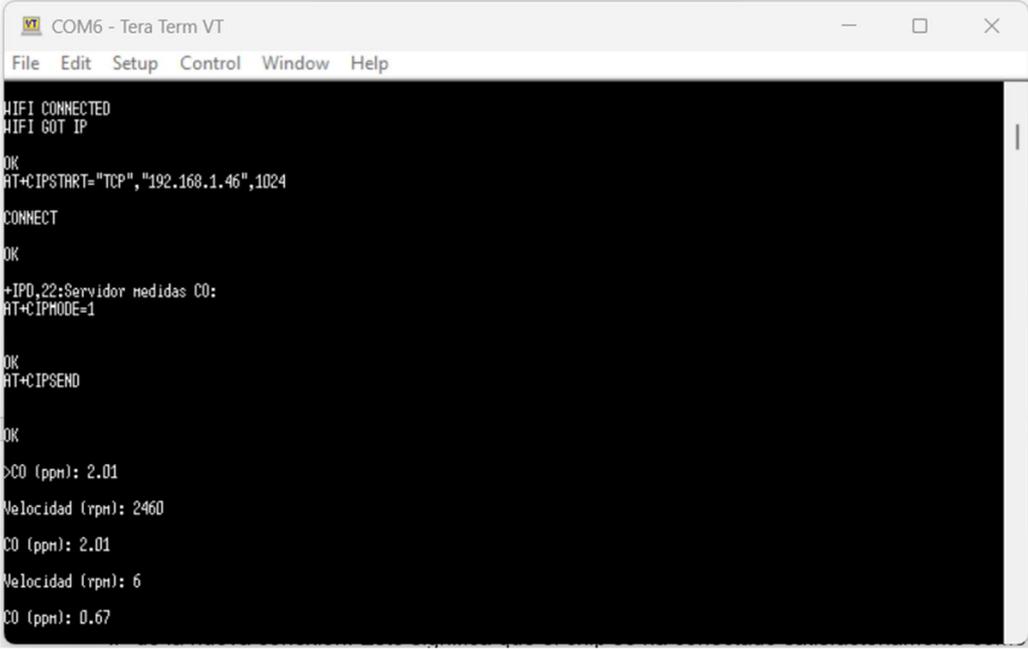


Figura 106. Prueba de funcionamiento: pestaña inicial de la aplicación, sin iniciar la comunicación

Para comenzar la ejecución del programa, se ha de encender el interruptor del módulo de fuente de alimentación y pulsar el botón negro de *reset* incluido en la placa. Al hacerlo, arranca el envío de comandos AT por el pin TX del microcontrolador. Los mensajes que se están mandando por protocolo UART al módulo wifi y las respuestas de este se pueden leer en tiempo real desde el terminal de Tera Term. Al enviar el comando “AT+CIPSTART”, se puede leer en el bloque de texto ‘Estado del servidor’ de la aplicación la IP de la nueva conexión (Figura 108). Esto significa que el chip se ha conectado satisfactoriamente como cliente.



```
COM6 - Tera Term VT
File Edit Setup Control Window Help
WIFI CONNECTED
WIFI GOT IP
OK
AT+CIPSTART="TCP","192.168.1.46",1024
CONNECT
OK
+IPD,22:Servidor medidas CO:
AT+CIPMODE=1
OK
AT+CIPSEND
OK
>CO (ppm): 2.01
Velocidad (rpm): 2460
CO (ppm): 2.01
Velocidad (rpm): 6
CO (ppm): 0.67
```

Figura 107. Prueba de funcionamiento: Terminal abierto en Tera Term y mensajes enviados por el microcontrolador

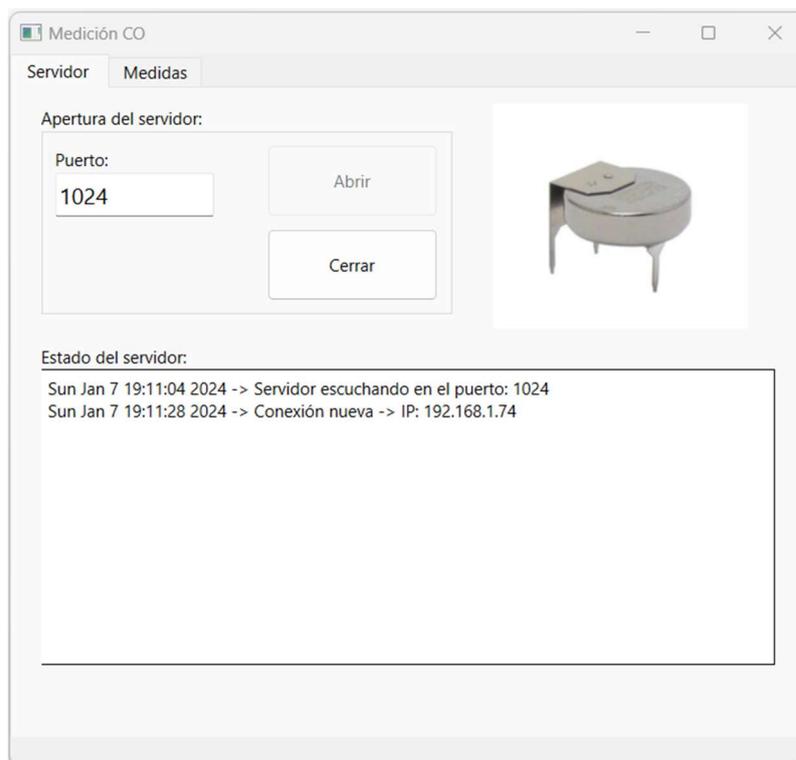


Figura 108. Prueba de funcionamiento: pestaña Servidor. Puerto abierto a la comunicación y conexión del chip wifi al servidor

Tras el envío del último comando, comienza la tarea periódica de toma y envío de medidas de CO en el aire y de velocidad de giro del *encoder*. Se ha programado para que esto suceda cada 15 segundos, de modo que se pueda ver una variación rápidamente.

En un estado de ausencia de monóxido de carbono, los valores medidos y mostrados deberían ser cercanos a 0 y el motor encontrarse parado. Para probar la respuesta ante un aumento de la concentración de este gas, se succiona el aire alrededor de la llama de un mechero con la bomba incluida con el sensor y se expulsa cerca del mismo. En cuanto se completa el ciclo de lectura y envío de datos, se puede comprobar, por un lado, que en la aplicación aparece el nuevo valor y en la gráfica se produce un pico, y, por el otro, que físicamente el motor ha empezado a girar, puesto que está diseñado para simular un ventilador que desplazaría este aire tóxico.

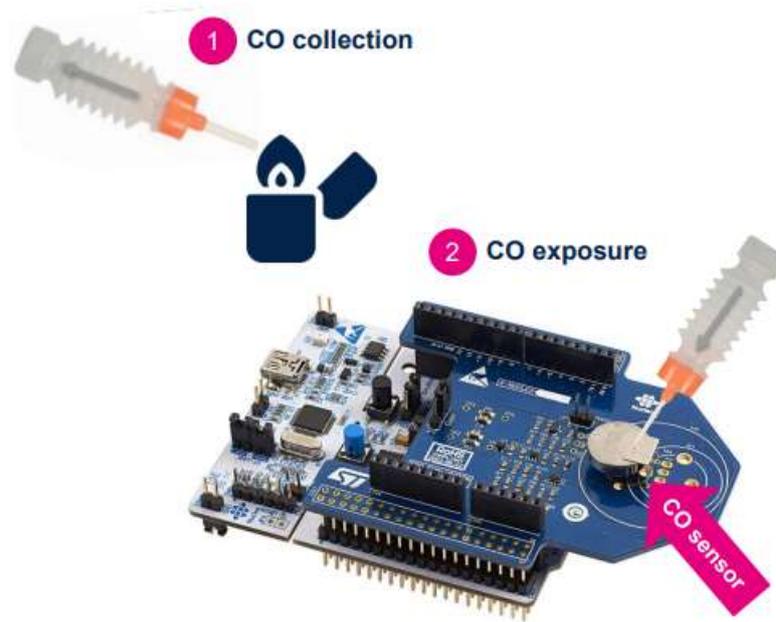


Figura 109. Proceso para probar niveles de CO altos en la placa P-NUCLEO-IKA02A1

Cada vez que llega una nueva medida, esta se muestra correctamente en el bloque de texto de la parte superior izquierda de la pantalla correspondiente, marcándose este en rojo a modo de alarma al superarse un límite de peligrosidad. Además, dicha medida se almacena en la base de datos, la cual se puede leer por pantalla, guardar en bloque de texto o borrar con los botones de la parte derecha.

Además, las mediciones de giro del encoder también aparecen en esta pantalla junto a su unidad de medida.

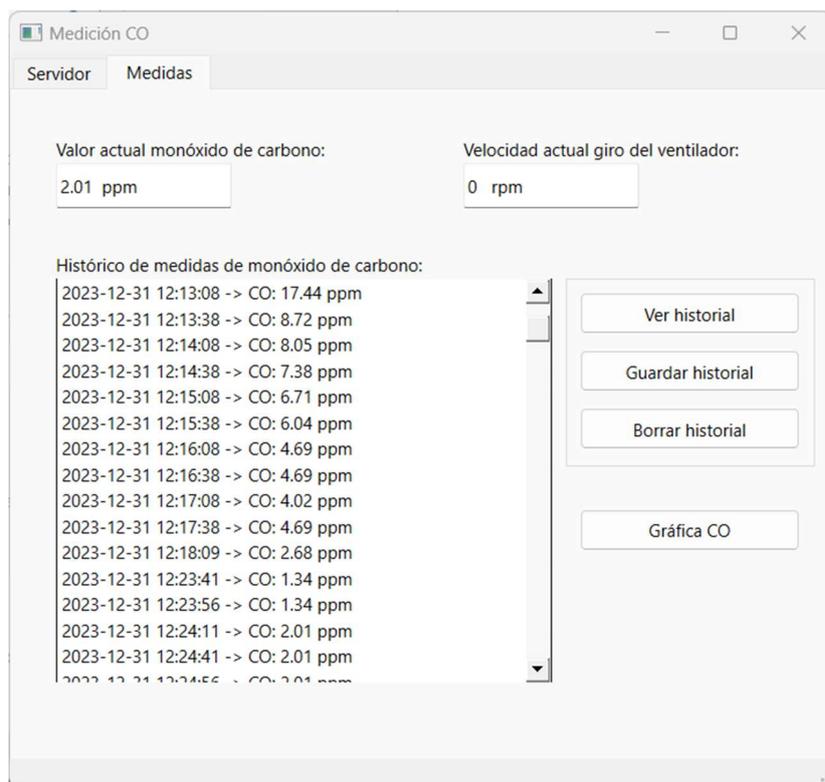


Figura 110. Prueba de funcionamiento: pestaña de Medidas cuando el monóxido de carbono detectado no supera el valor límite seguro

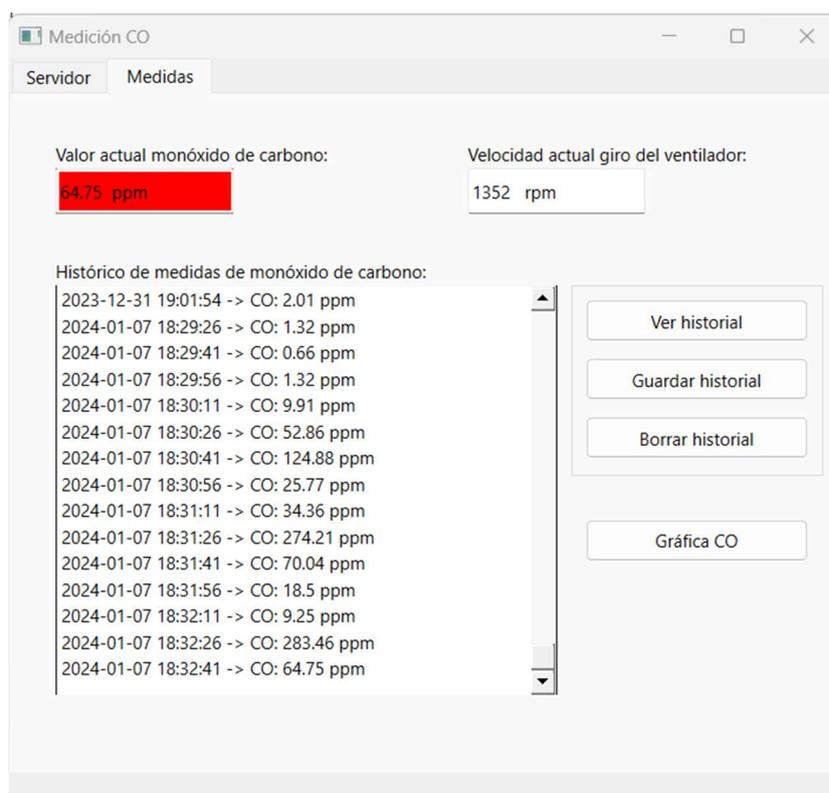


Figura 111. Prueba de funcionamiento: pestaña de Medidas cuando el monóxido de carbono detectado supera el valor límite seguro y se activan la alarma y el motor

Pulsando el botón 'Gráfica CO', se abre una ventana con una gráfica que muestra cómo varía la medida de CO con respecto al tiempo.

En la primera de ellas, se mantuvo varios minutos el dispositivo encendido. Se puede comprobar como la gráfica se mantiene cerca del 0 en todo momento excepto en los primeros segundos, el único momento en el que se generó monóxido de carbono con el mechero.

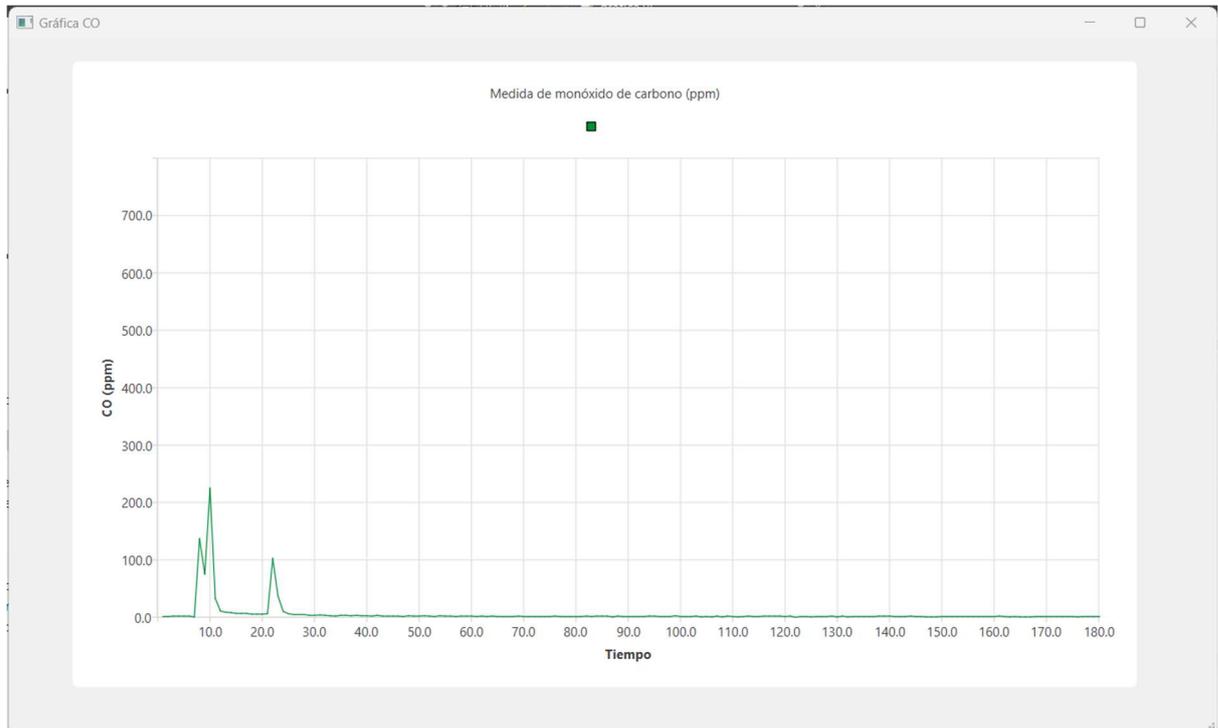


Figura 112. Prueba de funcionamiento: ventana con un gráfico que representa el valor medido de monóxido de carbono con respecto al tiempo

En la segunda gráfica se puede ver algo similar, pero en este caso se ha aproximado la vista. Hay tres picos que corresponden al periodo de tiempo en el que el sensor detectaba gas, y una vez dejaba de hacerlo, la función volvía a aproximarse al 0.

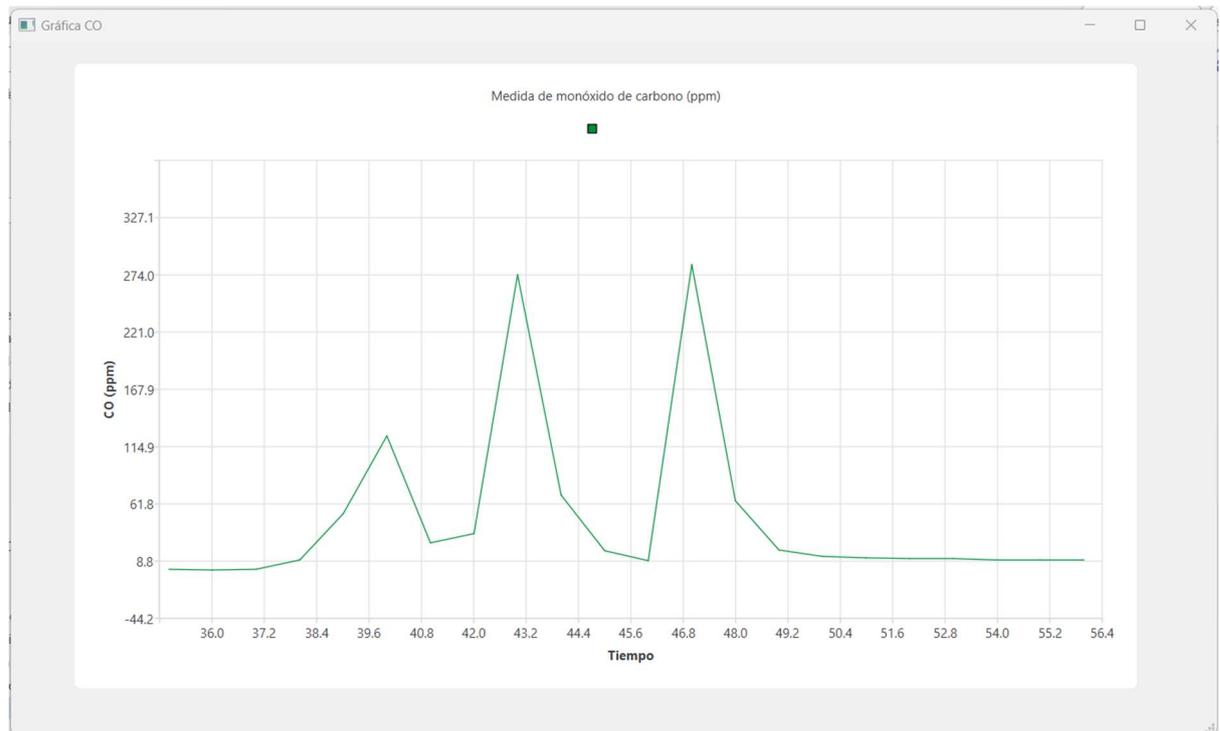


Figura 113. Prueba de funcionamiento: ventana con un gráfico que representa el valor medido de monóxido de carbono con respecto al tiempo (vista ampliada)

Por otro lado, relacionado con la base de datos, se puede comprobar que se genera correctamente el documento de texto con el historial de medidas cuando se pulsa el botón 'Guardar Historial' y que se guarda en la carpeta del proyecto.

```

MedidasCO.txt
Archivo  Editar  Ver
2023-12-31 11:59:06 -> CO: 4.02 ppm
2023-12-31 11:59:36 -> CO: 3.35 ppm
2023-12-31 12:00:06 -> CO: 4.02 ppm
2023-12-31 12:00:36 -> CO: 4.02 ppm
2023-12-31 12:01:06 -> CO: 2.68 ppm
2023-12-31 12:11:07 -> CO: 1.34 ppm
2023-12-31 12:11:37 -> CO: 1.34 ppm
2023-12-31 12:12:08 -> CO: 2.01 ppm
2023-12-31 12:12:38 -> CO: 339.35 ppm
2023-12-31 12:13:08 -> CO: 17.44 ppm
2023-12-31 12:13:38 -> CO: 8.72 ppm
2023-12-31 12:14:08 -> CO: 8.05 ppm
2023-12-31 12:14:38 -> CO: 7.38 ppm
2023-12-31 12:15:08 -> CO: 6.71 ppm
2023-12-31 12:15:38 -> CO: 6.04 ppm
2023-12-31 12:16:08 -> CO: 4.69 ppm
2023-12-31 12:16:38 -> CO: 4.69 ppm
2023-12-31 12:17:08 -> CO: 4.02 ppm
2023-12-31 12:17:38 -> CO: 4.69 ppm
2023-12-31 12:18:09 -> CO: 2.68 ppm
2023-12-31 12:23:41 -> CO: 1.34 ppm
2023-12-31 12:23:56 -> CO: 1.34 ppm
2023-12-31 12:24:11 -> CO: 2.01 ppm
2023-12-31 12:24:41 -> CO: 2.01 ppm
2023-12-31 12:24:56 -> CO: 2.01 ppm
2023-12-31 12:25:11 -> CO: 0.67 ppm
2023-12-31 12:25:26 -> CO: 136.81 ppm
Ln 1, Col 1      100%      Windows (CRLF)      UTF-8
    
```

En caso de que se quieran ver todos los pasos, el siguiente enlace lleva a un video con el funcionamiento de la aplicación y el terminal:

<https://youtu.be/3ivNHbK0fkk>

En definitiva, con esta prueba del proceso completo, se puede determinar que el dispositivo funciona de la manera que se había diseñado y se han cumplido los objetivos planteados: es capaz de tomar una medida correcta de CO, conectarse vía wifi a una aplicación donde el usuario puede consultarla, encender un motor y generar una señal PWM que controla la velocidad a la que gira dependiendo de la concentración ambiental de este gas, así como tener la posibilidad de leerla. Las interrupciones del temporizador 6, el botón azul y el índice del *encoder* rotatorio funcionan como se había planteado.

8 CONCLUSIONES

La primera conclusión es positiva y se obtiene observando los resultados de las pruebas: el dispositivo funciona de acuerdo con su diseño y realiza correctamente todas las tareas para las que estaba construido.

Al plantear sus características, se establecieron varias condiciones que debían acatarse para elaborar un aparato domótico integro, entre ellas la capacidad de medición de monóxido de carbono en el aire, el control de procesos mediante un microcontrolador, el tener la posibilidad de enviar información y mostrarla por pantalla y el de elaborar una maqueta para simular el funcionamiento de un ventilador real. Todos estos requisitos han sido satisfechos durante el desarrollo del trabajo, resultando en un dispositivo apto para cumplir con las necesidades planteadas.

En cuanto a la programación del software, se ha desarrollado un código que incluye una rutina de automatización y permite al usuario visualizar medidas de monóxido de carbono con un retardo mínimo.

Este trabajo se desarrolló con la idea de poner en práctica diversos aspectos de la electrónica, la electricidad y las telecomunicaciones, aplicándolos en el diseño y desarrollo de un dispositivo tangible, cuyo funcionamiento se pudiera evaluar de forma sencilla. En ese sentido, el proyecto ha cumplido por completo con su razón de ser.

Muchos dispositivos domóticos, como es el caso del desarrollado en este proyecto, tienen la finalidad de ayudar y proteger a las personas en sus hogares, por lo que se debe cuidar que funcionen correctamente. Cuando se trata de seguridad, es importante hacer las cosas de forma adecuada, incluso en este caso, en el que el producto no podría llegar al mercado debido al uso de los módulos de evaluación STM32, especialmente el que incluye el sensor de gas. Aun así, se ha tratado de elaborar un aparato funcional, completo y seguro.

Para la elección de los componentes se priorizó la sencillez en el montaje y configuración por encima de aspectos como el tamaño y el coste, sobre todo en lo que respecta a usar de placas de expansión y evaluación en vez de únicamente los componentes por separado. Esto se traduce en un dispositivo algo grande y tosco, pero cómodo a la hora de conectar los cables y componentes entre ellos. Como el objetivo final era crear una maqueta que simulara el funcionamiento se considera que, si bien no sería la solución ideal, sí es adecuada.

Por otro lado, el proyecto tiene una buena cantidad de posibilidades de ampliación. El hecho de que se haya construido de forma modular permite la adición o sustitución de placas de ST, colocando unas encima de otras dependiendo de las funciones que se quieran añadir y siempre que el uso de los pines sea compatible.

Asimismo, también hay cabida a proyectos de mejora, puesto que el dispositivo cuenta con una serie de limitaciones que se podrían sobrevenir dedicándole más tiempo. Entre ellas, se puede destacar el hecho de que la red wifi y el puerto del servidor al que se puede conectar están definidos por programa, lo cual restringe enormemente su portabilidad y rapidez de instalación. También, puesto que el aparato creado es solo una maqueta, existe la posibilidad de mejorar las conexiones y reducir el tamaño diseñando una placa de circuito impreso en vez de utilizar módulos de evaluación y cables tipo Arduino.

Para resumir, en este trabajo se ha logrado desarrollar un aparato con una considerable cantidad de módulos y funciones y, al hacerlo, se han puesto en práctica muchos conceptos diferentes de la electrónica: lectura de señales analógicas, transmisión por protocolo UART, comunicación inalámbrica, uso de temporizadores para la generación de señales PWM y para repetir tareas periódicas, control de la velocidad de rotación de un motor, desarrollo de interfaces gráficas, comunicación entre dispositivos por protocolo TCP/IP, programación de interrupciones del programa y la utilización de un codificador rotatorio para medir la velocidad de giro de un motor.

9 REFERENCIAS

- [1] IDAE, Instituto para la Diversificación y Ahorro de la Energía. Ministerio para la Transición Ecológica y el Reto Demográfico, «DOMÓTICA,» [En línea]. Available: <https://www.idae.es/tecnologias/eficiencia-energetica/edificacion/domotica>. [Último acceso: 26 abril 2023].
- [2] SEPAR, Sociedad Española de Neumología y Cirugía Torácica, «Los siete contaminantes más peligrosos para la salud,» 2019. [En línea]. Available: https://www.separ.es/sites/default/files/SEPAR_contaminacion.pdf. [Último acceso: 23 mayo 2023].
- [3] EPA, Agencia de Protección Ambiental de Estados Unidos, «Monóxido de Carbono,» diciembre 2022. [En línea]. Available: <https://espanol.epa.gov/cai/monoxido-de-carbono>. [Último acceso: 12 junio 2023].
- [4] Agencia EFE, «Las intoxicaciones por monóxido alcanzan su cifra más baja en diez años en Castilla y León,» 14 junio 2023. [En línea]. Available: <https://efe.com/castilla-y-leon/2023-06-14/las-intoxicaciones-por-monoxido-alcanzan-su-cifra-mas-baja-en-diez-anos-en-castilla-y-leon/>. [Último acceso: 08 septiembre 2023].
- [5] El Mundo, «Monóxido de carbono, el 'veneno silencioso',» 18 diciembre 2022. [En línea]. Available: <https://www.elmundo.es/comunidad-valenciana/castellon/2022/12/18/639e3332fdddffcf808b456d.html>. [Último acceso: 08 septiembre 2023].
- [6] SEPAR, Sociedad Española de Neumología y Cirugía Torácica, «La contaminación del aire causa 10.000 muertes al año en España, muchas más que las 1.700 por accidentes de tráfico,» 04 junio 2019. [En línea]. Available: <https://www.separ.es/sites/default/files/SEPAR%20NP%20medio%20ambiente%20y%20salud%20%284%20jun%2019%29.pdf>. [Último acceso: 08 septiembre 2023].
- [7] BOE, Boletín Oficial del Estado, «Real Decreto 102/2011, de 28 de enero, relativo a la mejora de la calidad del aire,» enero 2011. [En línea]. Available: <https://www.boe.es/eli/es/rd/2011/01/28/102>. [Último acceso: 12 junio 2023].
- [8] ATSDR, Agencia para Sustancias Tóxicas y el Registro de Enfermedades de Estados Unidos, «Resúmenes de Salud Pública - Monóxido de Carbono (Carbon Monoxide),»

- mayo 2016. [En línea]. Available: https://www.atsdr.cdc.gov/es/phs/es_phs201.html. [Último acceso: 12 junio 2023].
- [9] Murcia Salud, portal sanitario de la Región de Murcia, «Monóxido de carbono,» [En línea]. Available: <https://www.murciasalud.es/pagina.php?id=180398&idsec=1573#:~:text=El%20mon%C3%B3xido%20de%20carbono%2C%20cuya,en%20ambientes%20de%20poco%20ox%C3%ADgeno..> [Último acceso: 12 junio 2023].
- [10] HANWEI ELECTRONICS CO., LTD, «TECHNICAL DATA MQ-7 GAS SENSOR,» [En línea]. Available: <https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-7.pdf>. [Último acceso: 28 junio 2023].
- [11] HANWEI ELECTRONICS CO., LTD, «TECHNICAL DATA MQ-9 GAS SENSOR,» 2023. [En línea]. Available: https://www.electronicoscaldas.com/datasheet/MQ-9_Hanwei.pdf. [Último acceso: 30 junio 2023].
- [12] Figaro USA inc., «TGS5141-P00 Carbon Monoxide Sensor,» [En línea]. Available: <https://www.figarosensor.com/product/entry/tgs5141-p00.html>. [Último acceso: 2023 junio 15].
- [13] Figaro USA inc., «TGS 5141-P00 for the detection of Carbon Monoxide. Datasheet,» noviembre 2018. [En línea]. Available: https://www.figaro.co.jp/en/product/docs/tgs5141-p00_product%20information%28en%29_rev05.pdf. [Último acceso: 12 noviembre 2023].
- [14] SemTech, «What is LoRa?,» [En línea]. Available: <https://www.semtech.com/lora>. [Último acceso: 2023 junio 25].
- [15] Lora Alliance, «What is LoRaWan Specification,» [En línea]. Available: <https://loralliance.org/about-lorawan/>. [Último acceso: 25 junio 2023].
- [16] Bluetooth, «Learn about Bluetooth,» [En línea]. Available: <https://www.bluetooth.com/learn-about-bluetooth/>. [Último acceso: 28 junio 2023].
- [17] À. Perles, «1 - Introducción a los microcontroladores STM32 de ST. UPV.,» [En línea]. Available: https://www.disca.upv.es/aperles/arm_cortex_m3/libre/libro-ARM-Cortex-M.pdf. [Último acceso: 19 9 2023].

- [18] ST, «Arm® Cortex®-M in a nutshell,» [En línea]. Available: https://www.st.com/content/st_com/en/arm-32-bit-microcontrollers.html. [Último acceso: 15 mayo 2023].
- [19] ebmpapst, «W3G300-UL11-35 EC axial fan datasheet,» [En línea]. Available: <https://img.ebmpapst.com/products/manuals/W3G300UL1135-BA-ENU.pdf>. [Último acceso: 16 agosto 2023].
- [20] Nidec, «DMN29 Motor datasheet,» [En línea]. Available: <https://docs.rs-online.com/cbef/0900766b81443008.pdf>. [Último acceso: 19 agosto 2023].
- [21] ST, «P-NUCLEO-IKA02A1,» [En línea]. Available: <https://www.st.com/en/ecosystems/p-nucleo-ika02a1.html>. [Último acceso: 03 julio 2023].
- [22] ST, «STLM20 datasheet,» [En línea]. Available: <https://www.st.com/resource/en/datasheet/stlm20.pdf>. [Último acceso: agosto 2023].
- [23] ST, «STM32L053C6 STM32L053C8 STM32L053R6 STM32L053R8 datasheet,» [En línea]. Available: <https://www.st.com/resource/en/datasheet/stm32l053c6.pdf>. [Último acceso: 15 noviembre 2023].
- [24] Arm, «About μ Vision,» [En línea]. Available: <https://developer.arm.com/documentation/101407/0538/About-uVision>. [Último acceso: 17 octubre 2023].
- [25] Espressif Systems, «ESP8266 AT Instruction Set. Version 1.5.4,» [En línea]. Available: https://www.espressif.com/sites/default/files/4a-esp8266_at_instruction_set_en_v1.5.4_0.pdf. [Último acceso: 10 octubre 2023].
- [26] Espressif Systems, «ESP8266 AT Command Examples. Version 1.3,» [En línea]. Available: https://www.espressif.com/sites/default/files/4b-esp8266_at_command_examples_en_v1.3.pdf. [Último acceso: 18 octubre 2023].
- [27] IBM, «Protocolos TCP/IP,» marzo 2021. [En línea]. Available: <https://www.ibm.com/docs/es/aix/7.1?topic=protocol-tcpip-protocols>. [Último acceso: 20 octubre 2023].
- [28] AVG, «¿Qué es TCP/IP?,» junio 2021. [En línea]. Available: <https://www.avg.com/es/signal/what-is-tcp-ip>. [Último acceso: 18 octubre 2023].

- [29] QT Group, «Qt Creator,» [En línea]. Available: <https://www.qt.io/product/development-tools>. [Último acceso: 26 octubre 2023].
- [30] DISCA UPV, *Presentaciones asignatura Sistemas Informáticos Industriales (SII 12164)*, Valencia: Universitat Politècnica de València, 2021-2022.
- [31] QT, «Database Classes,» 2023. [En línea]. Available: <https://doc.qt.io/qt-6/database.html>. [Último acceso: 05 noviembre 2023].
- [32] ST, «STSPIN250 datasheet Rev 2,» noviembre 2016. [En línea]. Available: <https://www.st.com/resource/en/datasheet/stspin250.pdf>. [Último acceso: 09 noviembre 2023].
- [33] ST, «X-NUCLEO-IHM13A1 datasheet Rev 1,» diciembre 2016. [En línea]. Available: https://www.st.com/resource/en/data_brief/x-nucleo-ihm13a1.pdf. [Último acceso: 09 noviembre 2023].
- [34] ST, «Getting started with the P-NUCLEO-IKA02A1 STM32 Nucleo pack for electrochemical toxic gas sensor expansion board with CO sensor,» 2017. [En línea]. Available: https://www.st.com/resource/en/user_manual/um2247-getting-started-with-the-pnucleoika02a1-stm32-nucleo-pack-for-electrochemical-toxic-gas-sensor-expansion-board-with-co-sensor-stmicroelectronics.pdf. [Último acceso: 19 mayo 2023].
- [35] Figaro Inc, «APPLICATION NOTES FOR TGS5xxx SERIES. Datasheet,» [En línea]. Available: [https://www.figarosensor.com/product/docs/tgs5xxx_application%20note\(en\)_rev01.pdf](https://www.figarosensor.com/product/docs/tgs5xxx_application%20note(en)_rev01.pdf). [Último acceso: 12 noviembre 2023].
- [36] Espressif Systems, «ESP8266EX datasheet Rev 7.0,» junio 2023. [En línea]. Available: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf. [Último acceso: 16 septiembre 2023].
- [37] ST, «STM32L4 - Timers,» [En línea]. Available: https://www.st.com/resource/en/product_training/STM32L4_WDG_TIMERS_GPTIM.pdf. [Último acceso: 06 11 2023].

- [38] CUI Devices, «AMT10/10E Encoder Assembly Instructions | CUI Devices,» [En línea]. Available: <https://www.cuidevices.com/catalog/resource/amt10-10e-assembly-instructions.pdf>. [Último acceso: 12 12 2023].
- [39] CUI Devices, «AMT102-V Datasheet - Modular | Incremental | CUI Devices,» 2022. [En línea]. Available: <https://www.cuidevices.com/product/resource/amt10-v.pdf>. [Último acceso: 11 12 2023].
- [40] Analog Devices, «UART: A Hardware Communication Protocol Understanding Universal Asynchronous Receiver/Transmitter,» [En línea]. [Último acceso: 01 septiembre 2023].
- [41] ST, «Getting started with UART,» [En línea]. Available: https://wiki.st.com/stm32mcu/wiki/Getting_started_with_UART. [Último acceso: 06 septiembre 2023].
- [42] ST, «X-NUCLEO-IHM13A1,» [En línea]. Available: <https://www.st.com/en/ecosystems/x-nucleo-ihm13a1.html>. [Último acceso: 09 noviembre 2023].
- [43] À. Perles Ivars, «Interrupciones con el periférico EXTI en los microcontroladores STM32F4 de la arquitectura ARM Cortex-M,» [En línea]. Available: <https://riunet.upv.es/bitstream/handle/10251/52760/Interrupciones%20EXTI%20en%20microcontroladores%20STM32F4.pdf>. [Último acceso: 20 12 2023].
- [44] Naciones Unidas, «Objetivos de Desarrollo Sostenible. Portada.,» [En línea]. Available: <https://www.un.org/sustainabledevelopment/es/>. [Último acceso: 04 enero 2024].
- [45] Naciones Unidas, «ODS 3: Salud y Bienestar,» [En línea]. Available: <https://www.un.org/sustainabledevelopment/es/health/>. [Último acceso: 04 enero 2024].
- [46] Naciones Unidas, «ODS 9: Industria, Innovación e Infraestructuras,» [En línea]. Available: <https://www.un.org/sustainabledevelopment/es/infrastructure/>. [Último acceso: 04 enero 2024].
- [47] Naciones Unidas, «ODS 11: Ciudades y Comunidades Sostenibles,» [En línea]. Available: <https://www.un.org/sustainabledevelopment/es/cities/>. [Último acceso: 04 enero 2024].

ANEXO I: OBJETIVOS DE DESARROLLO SOSTENIBLE

En esta sección se discute la relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS) de la agenda 2030.

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.			X	
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.	X			
ODS 4. Educación de calidad.				X
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.			X	
ODS 7. Energía asequible y no contaminante.		X		
ODS 8. Trabajo decente y crecimiento económico.			X	
ODS 9. Industria, innovación e infraestructuras.	X			
ODS 10. Reducción de las desigualdades.		X		
ODS 11. Ciudades y comunidades sostenibles.	X			
ODS 12. Producción y consumo responsables.		X		
ODS 13. Acción por el clima.		X		
ODS 14. Vida submarina.			X	
ODS 15. Vida de ecosistemas terrestres.		X		
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos.				X

Tabla 18. Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Los Objetivos de Desarrollo Sostenible (ODS) son un conjunto de iniciativas propuestas por las Naciones Unidas para luchar contra la pobreza mundial y el cambio climático y para garantizar que el planeta pueda disfrutar de paz y prosperidad para el año 2030. De los 17 objetivos planteados, este proyecto tiene, sobre todo, relación con aquellos centrados en mejorar la calidad de vida de las personas, especialmente los 3 siguientes:

- Objetivo 3. Salud y bienestar: este objetivo está centrado en garantizar la salud de las personas y promover un estilo de vida saludable.

El monóxido de carbono es un gas que, al ser inhalado, llega a sustituir al oxígeno que lleva la sangre provocando que este no llegue a los distintos tejidos del cuerpo y,

de este modo, resultando en problemas de salud graves a corto y largo plazo e incluso la muerte.

Incorporar en un espacio cerrado un dispositivo que pueda detectar cuando los niveles de este gas pueden ser perjudiciales para las personas contribuye de forma directa a prevenir intoxicaciones, evitando de esta manera enfermedades y muertes.

El uso de este tipo de aparatos permite crear entornos más seguros y saludables, por lo que se alinea con el ODS 3 y su premisa de salud global y de reducción de muertes y enfermedades producidas, entre otros, por contaminantes en el aire.

- Objetivo 9. Industria, innovación y estructuras: la iniciativa 9 está centrada en la construcción de infraestructuras de calidad, la industrialización sostenible y la innovación en la tecnología y su accesibilidad para todas las personas.

Tomando esto en consideración, el desarrollo de un dispositivo con conexión inalámbrica y la capacidad de detectar el CO en el aire mediante sensores avanzados es un caso lógico de la aplicación de la tecnología y la innovación para mejorar la vida de las personas.

Además, la instalación de este aparato en un edificio contribuiría a hacerlo más seguro, protegiendo a las personas que lo ocupan de un posible escape de este gas que, de otra forma, sería difícil de detectar al no tener color ni olor.

- ODS 11: Ciudades y Comunidades Sostenibles: el undécimo objetivo está enfocado en la mejora de las ciudades para hacerlas más eficientes, accesibles y sostenibles. Para hacerlo, la seguridad en los edificios y hogares es un aspecto indispensable.

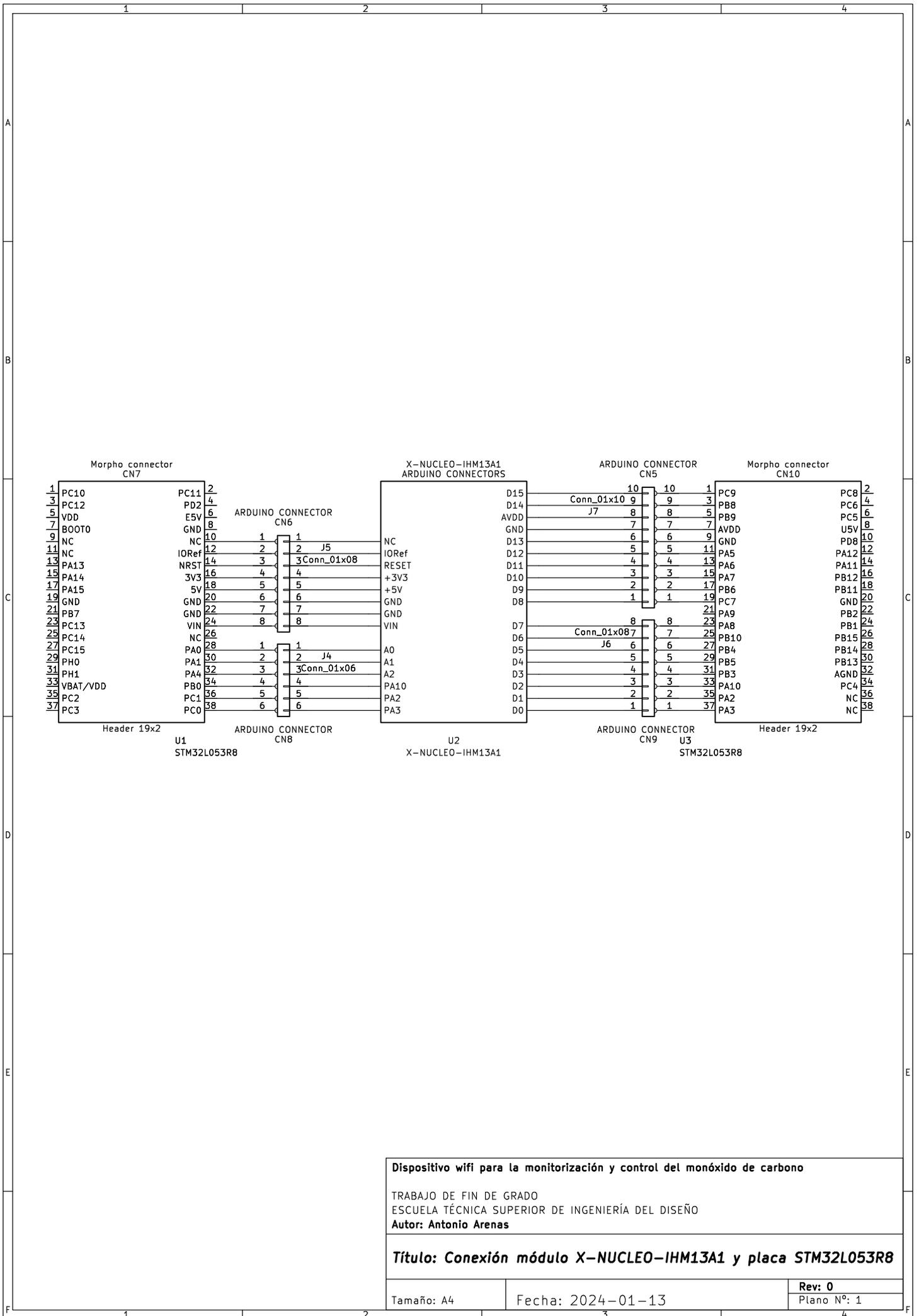
Por tanto, el desarrollo de este dispositivo, que es capaz de detectar y avisar cuando el nivel de un gas tóxico supera los límites saludables, contribuye a crear interiores más seguros y a aumentar el bienestar en las ciudades.

Además, un número considerable de intoxicaciones por monóxido de carbono ocurre a causa de estufas antiguas o en mal estado, de este modo, la instalación de un detector de CO mejoraría la calidad de vida y evitaría desgracias en los hogares más desfavorecidos, donde este tipo de aparatos de gas son más comunes.

2. PLANOS

Índice Planos

1	CONEXIÓN MÓDULO X-NUCLEO-IHM13A1 Y PLACA STM32L053R8.....	151
2	CONEXIÓN MÓDULOS P-NUCLEO-IKA02A1 Y X-NUCLEO-IHM13A1.....	153
3	DIAGRAMA ELECTRÓNICO.....	155

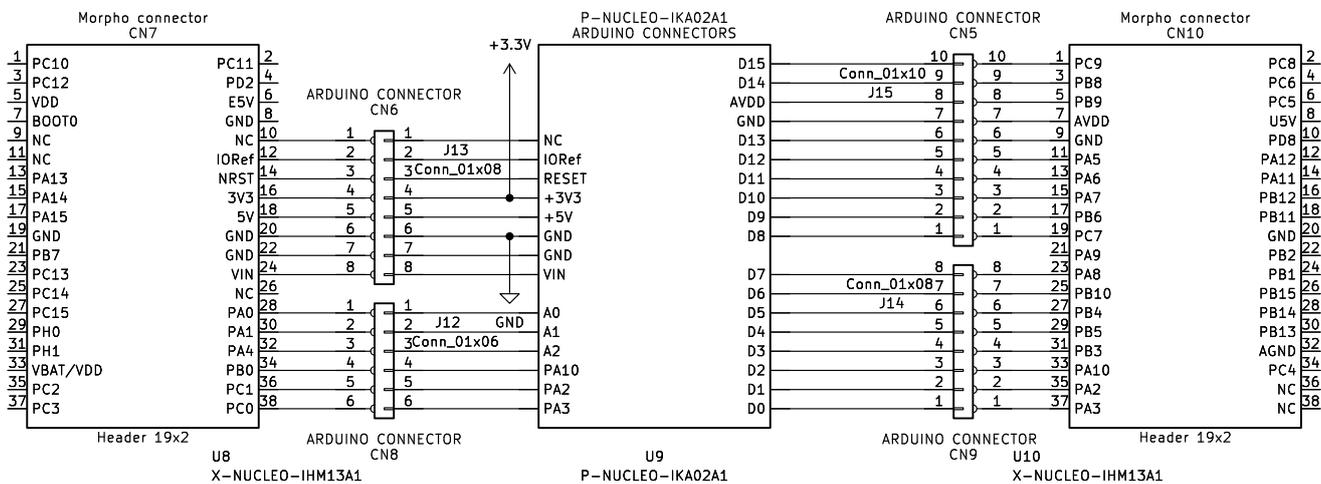


Dispositivo wifi para la monitorización y control del monóxido de carbono

TRABAJO DE FIN DE GRADO
 ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DEL DISEÑO
 Autor: Antonio Arenas

Título: Conexión módulo X-NUCLEO-IHM13A1 y placa STM32L053R8

Tamaño: A4	Fecha: 2024-01-13	Rev: 0 Plano N°: 1
------------	-------------------	-----------------------



Dispositivo wifi para la monitorización y control del monóxido de carbono

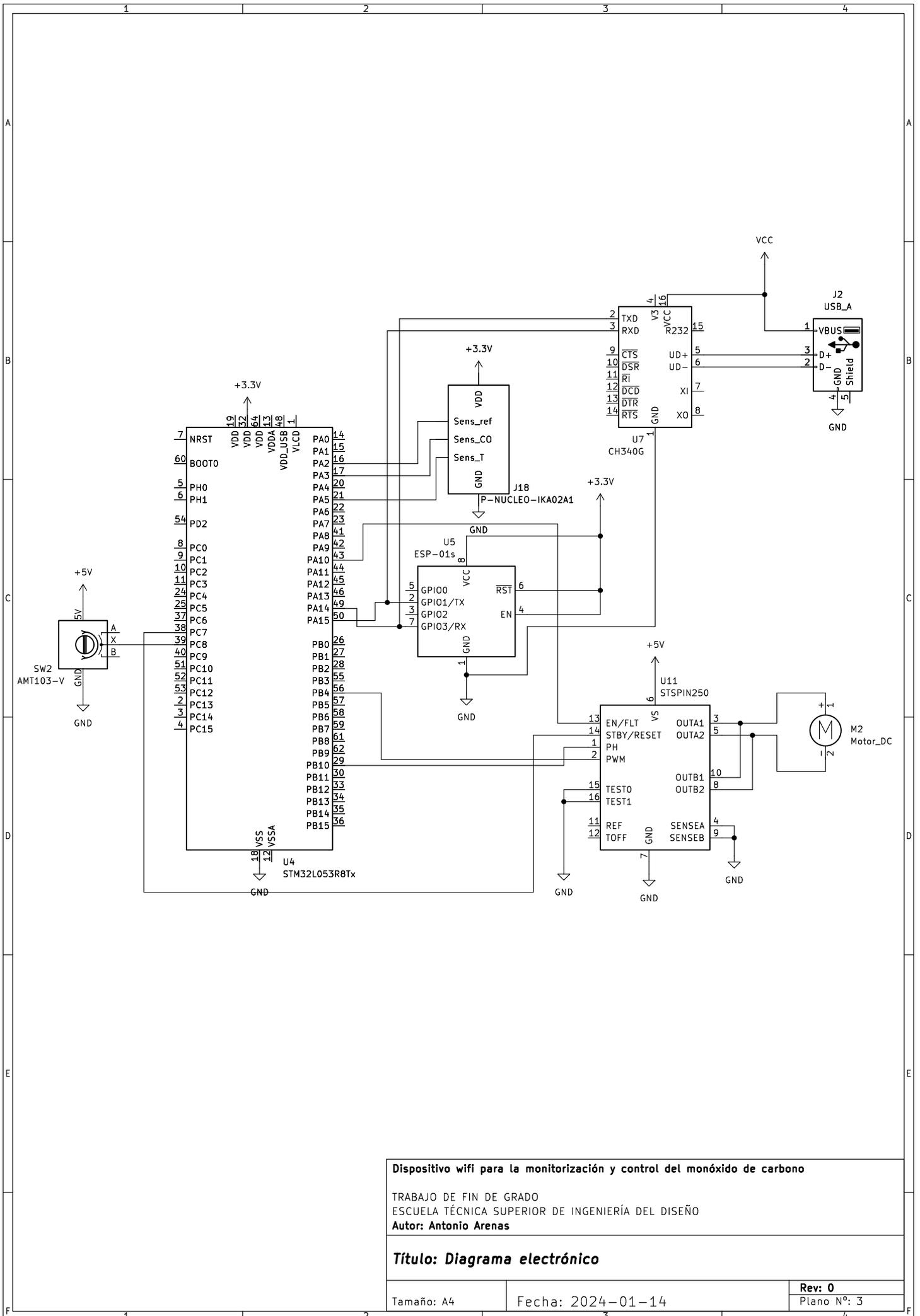
TRABAJO DE FIN DE GRADO
 ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DEL DISEÑO
 Autor: Antonio Arenas

Título: Conexión módulos P-NUCLEO-IKA02A1 y X-NUCLEO-IHM13A1

Tamaño: A4

Fecha: 2024-01-13

Rev: 0
 Plano N°: 2



Dispositivo wifi para la monitorización y control del monóxido de carbono

TRABAJO DE FIN DE GRADO
 ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DEL DISEÑO
 Autor: Antonio Arenas

Título: Diagrama electrónico

Tamaño: A4	Fecha: 2024-01-14	Rev: 0
		Plano N°: 3

3. PLIEGO DE CONDICIONES

Índice Pliego de condiciones

1	OBJETO	159
2	NORMATIVA DE CARÁCTER GENERAL.....	159
3	MATERIALES	159
3.1	ELECTRÓNICA	159
3.1.1	<i>Módulos.....</i>	<i>159</i>
3.2	ALIMENTACIÓN	161
3.2.1	<i>Módulo de fuente de alimentación</i>	<i>161</i>
3.2.2	<i>Adaptador convertidor AC/DC de 9 V, 1 A.....</i>	<i>161</i>
3.2.3	<i>Cableado.....</i>	<i>161</i>
3.3	ELEMENTOS ELECTROMECÁNICOS	161
3.3.1	<i>Motor DC.....</i>	<i>161</i>
3.3.2	<i>Codificador rotatorio AMT103-V.....</i>	<i>161</i>
3.4	CONTROL DE CALIDAD.....	162
4	CONDICIONES DE LA EJECUCIÓN	162
4.1	MONTAJE	162
5	PRUEBAS DE SERVICIO	163
5.1	PRUEBA FUENTE DE ALIMENTACIÓN	163
5.2	PRUEBA INTERRUPTOR	164
5.3	PRUEBA DE MEDICIÓN DE MONÓXIDO DE CARBONO.	164
5.4	PRUEBA DE CONEXIÓN WIFI	164
5.5	PRUEBA GIRO DEL MOTOR.....	165
5.6	PRUEBA CONEXIÓN A SERVIDOR	165
6	PRUEBA GENERAL.....	166

1 OBJETO

La presente especificación se refiere al conjunto de trabajos necesarios para la construcción e implementación de un dispositivo electrónico para la monitorización y control del monóxido de carbono, con conectividad wifi y capacidad de control de un motor.

2 NORMATIVA DE CARÁCTER GENERAL

El proyecto deberá realizarse acogiéndose a la siguientes normas y directivas:

- **UNE-EN 54-31:2014+A1:2019.** Sistemas de detección y alarma de incendios. Parte 31: Detectores de incendios multisensoriales. Detectores puntuales que utilizan una combinación de humo, monóxido de carbono y opcionalmente sensores de calor.
- **UNE-EN 50291-1:2018 (Ratificada).** Detectores de gas. Aparatos eléctricos para la detección de monóxido de carbono en los locales de uso doméstico. Parte 1: Métodos de ensayo y requisitos de funcionamiento. (Ratificada por la Asociación Española de Normalización en junio de 2018.)
- **Directiva de Baja Tensión (2014/35/UE):** Se establecen unos requisitos mínimos de seguridad eléctrica para los productos de baja tensión que se comercializan en la Unión Europea.
- **Directiva Red (2014/53/UE):** Se determinan los requisitos mínimos para la comercialización de equipos radioeléctricos en la Unión Europea.

3 MATERIALES

3.1 Electrónica

3.1.1 Módulos

3.1.1.1 STM32L053R8

El control se realizará mediante el módulo STM32L053R8, que incluye el microcontrolador STM32L0 de bajo consumo basado en el ARM Cortex M0+ de 32 bits.

Este módulo deberá estar alimentado entre los 1.65 V a 3.6 V para su funcionamiento y deberá contar con capacidades de comunicación UART. Sus 51 pines GPIO deberán ser compatibles con conectores estándar Arduino o *ST morpho Connectors*. Deberá contar con un convertidor analógico-digital y otro digital-analógico de 12 bits.

3.1.1.2 P-NUCLEO-IKA02A1

La toma de medidas de monóxido de carbono se realizará mediante la placa P-NUCLEO-IKA02A1, que contiene el sensor de CO de alta precisión TGS5141.

Dicho sensor habrá de tener la capacidad de medir concentraciones de monóxido de carbono entre las 0 y las 5000 ppm, y contará con un valor de salida de corriente no inferior a los 1,2 nA/ppm y no superior a los 3,2 nA/ppm.

La placa P-NUCLEO-IKA02A1 deberá tener la capacidad de ser conectada al módulo STM32L053R8 e incluirá un sensor de temperatura STLM20. Este no deberá de tener un error superior a los $\pm 1,5$ °C a temperatura ambiente ni superior a los $\pm 2,5$ °C en los extremos de su rango de temperatura. El sensor no deberá estar alimentado a menos de 2,4 V ni a más de 5,5 V.

3.1.1.3 ESP-01s

El módulo ESP-01s, que incorpora el chip ESP8266EX, se utilizará para otorgar funcionalidades wifi al aparato. Contará con 8 pines para la alimentación y conexión con el microcontrolador. Deberá estar conectado a una fuente que genere una corriente no inferior a los 200 mA y una tensión no menor a 3,3 V y no superior a 3,6. El chip, además, deberá tener pre-instalado el firmware AT para poder recibir comandos de control por sus pines UART.

3.1.1.4 X-NUCLEO-IHM13A1

Con el objetivo de realizar el control de la velocidad del motor DC se utilizará el *driver* STSPIN250 integrado en la placa X-NUCLEO-IHM13A1. Este deberá recibir una señal PWM del microcontrolador que luego adaptará para alimentar el motor. El módulo habrá de estar abastecido mediante un voltaje no inferior a 1,8 V y no superior a 10 V y proporcionar una corriente eficaz no superior a 2,6 A RMS. Los leds incluidos en la placa deberán iluminarse cuando circule corriente a través de ellos.

3.1.1.5 Módulo USB-TTL CH340G

Se utilizará el módulo USB-TTL CH340G para poder leer los mensajes enviados por protocolo UART entre el microcontrolador y el módulo wifi en un pc. Este deberá tener necesariamente conexión USB tipo A en un lado y en otro tres conexiones: masa, recepción y transmisión.

3.2 Alimentación

3.2.1 Módulo de fuente de alimentación

Para proporcionar suficiente intensidad a los elementos del sistema se utilizará un módulo de fuente de alimentación conectado a una placa de prototipado. Este deberá suministrar 3,3 y 5 V en cada una de sus salidas y estar alimentado mediante una tensión no inferior a 6,5 V y no superior a 9 V. El botón que incorpora deberá ser capaz de impedir el paso de corriente en ambos pines de salida.

3.2.2 Adaptador convertidor AC/DC de 9 V, 1 A

Se utilizará un adaptador para convertir la corriente y tensión tomadas de la red en valores adecuados para alimentar el dispositivo. Este deberá incluir un led que se ilumine cuando circule corriente, que no deberá ser superior a 1 A. La tensión que proporciona no deberá superar los 9 V.

3.2.3 Cableado

Para el conexionado de los diferentes componentes electrónicos se usarán cables Dupont hembra/hembra de 20 cm de longitud y cables jumper macho/macho de diferentes longitudes, todos recubiertos de plástico ABS.

Para la escritura del programa desde el ordenador al microcontrolador se empleará un cable de tipo USB tipo A - mini USB.

3.3 Elementos electromecánicos

3.3.1 Motor DC

Se utilizará un motor de corriente continua de 12 V para simular el funcionamiento de un ventilador controlado por señales PWM. Este deberá poder girar cuando está alimentado a 5 V. Su velocidad de giro sin carga no deberá ser superior a 5000 revoluciones por minuto con un consumo de 0,07 A y deberá tener un eje rotatorio de diámetro no superior a 2,5 mm.

3.3.2 Codificador rotatorio AMT103-V

Para medir la velocidad de giro del motor DC se acoplará un *encoder* incremental en el extremo de su eje. Este deberá tener una resolución no inferior a 48 pulsos revolución y no superior a 2048. Además, no deberá utilizarse para medir velocidades superiores a 15000 rpm.

Puesto que se debe colocar alrededor del eje del motor, deberá incluir un manguito para ejes de 3mm diámetro.

3.4 Control de calidad

En cuanto al control de calidad de los materiales y componentes electrónicos listados, es responsabilidad del fabricante llevar a cabo los procesos pertinentes para asegurar la aptitud de los productos que comercializan. Aun así, se deberán inspeccionar visualmente previo al montaje para comprobar su integridad y buscar desperfectos superficiales.

Las pruebas de funcionamiento de los elementos electromecánicos también las deben realizar las distintas empresas fabricantes, comprobando, entre otras cosas, la seguridad y capacidad de giro del motor y la rotación del engranaje interno del *encoder*.

Además, todos los componentes listados deberán cumplir con la directiva europea 2011/65/UE de Restricción de ciertas Sustancias Peligrosas en aparatos eléctricos y electrónicos (RoHS).

4 CONDICIONES DE LA EJECUCIÓN

4.1 Montaje

Se utilizará una placa de prototipado para realizar el conexionado. Esta estará alimentada en sus carriles extremos por el módulo de fuente de alimentación, que deberá tener uno de sus jumpers cortocircuitando el pin de 5 V y el de VCC y el otro cortocircuitando el pin de 3,3 V y el de VCC. Por el momento, este módulo deberá estar desconectado de la red eléctrica.

Se deberá utilizar un destornillador con una punta plana de menos de tres mm para aflojar los tornillos de los 4 conectores de alimentación externos del módulo X-NUCLEO-IHM13A1, y así poder insertar cables macho-macho en su interior, 4 en total, asegurándose del contacto con las placas de metal. A continuación se deben volver a apretar estos tornillos.

Seguidamente, para el suministro energético al controlador de motores STSPIN250, se conectarán dos de estos cables, los introducidos en los conectores marcados como GND y Vin, en el carril de masa y en el de 5 V de la *proto-board*, respectivamente.

A continuación, se montará el módulo de control de motores (X-NUCLEO-IHM13A1) sobre la placa STM32L053R8, cuadrando los pines inferiores de la primera con los conectores ST *morpho* de la segunda. A continuación, se conectará de la misma forma el módulo de medición de monóxido de carbono (P-NUCLEO-IKA02A1) sobre el anterior.

En este módulo, además, habrá que ajustar la ganancia cortocircuitando los pines 2 y 3 con el *jumper* JP2.

Para alimentar el conjunto de aparatos electrónicos, se conectará un cable macho-macho entre el carril de 3,3 V de la *protoboard* y el pin tipo Arduino del canal 6 del módulo de CO marcado como 3V3. Asimismo, se conectará el pin GND del mismo canal con el carril de masa de la placa de prototipado.

Al montar el encoder AMT103-V sobre el motor, será deberán seguir todos los pasos indicados por el fabricante en el manual de montaje. Será preciso utilizar únicamente el manguito naranja para ejes de 3mm y dejar el interruptor DIP en la configuración por defecto (todos los conmutadores en estado de *off*).

Una vez se ha acoplado al motor, se conectará a los cables enchufados al módulo de control de motores para alimentarlo. La salida A+ se unirá con el terminal rojo del motor y la marcada como A- con el negro.

El módulo wifi ESP01s tendrá conexión en 6 de sus 8 pines. Puesto que son conectores macho, se utilizarán cables *dupont* macho-hembra para el conexionado. Este se realizará del siguiente modo: el pin marcado como 3V3 se enlazará con el bus de 3,3 V de la *protoboard*, igual que el pin EN (Enable) y el pin RST. El pin GND se unirá al bus de masa de la placa de pruebas. Por último, se van a asignar dos pistas de la placa de pruebas para la transmisión por protocolo UART. En una de ellas se conectará el pin de transmisión TX de la placa STM32L053R8 (A14) en serie con el pin de recepción RX del módulo y en otra el pin de recepción RX (A15) de la placa con el de transmisión TX del ESP01s.

El módulo USB-TTL CH340G también irá conectado a las pistas anteriormente mencionadas. El pin de transmisión TXD se conectará a la que tiene conectadas los pines de transmisión de la placa y de recepción del módulo wifi y el pin de recepción RXD se conectará a la otra. GND se acoplará al bus de masa de la *protoboard*. La conexión USB tipo A del módulo se enchufará a una entrada USB de un pc.

5 PRUEBAS DE SERVICIO

Se detallan las pruebas que se deben realizar para asegurar que el sistema y cada uno de sus componentes funcionan correctamente una vez montado el dispositivo.

5.1 Prueba fuente de alimentación

Se debe comprobar que el sistema está correctamente alimentado. Se conectará el convertidor AC/DC de 9 V a un enchufe y se comprobará que se enciende un led rojo al hacerlo. Con un multímetro se medirá la tensión entre el positivo de cada uno de los buses y masa para comprobar que se obtiene una tensión de 3,3 V y otra de 5 V.

Se dejará el dispositivo conectado durante un tiempo no inferior a 3 minutos, con el microcontrolador en estado de reposo, sin ejecutar ningún programa, de modo que el consumo sea uniforme y bajo durante la duración de la prueba.

5.2 Prueba interruptor

Es importante que exista un elemento que corte el paso de corriente al dispositivo a modo de seguridad. En este caso se utilizará el interruptor incluido en el módulo de fuente de alimentación.

Para la prueba, se pulsará el interruptor de modo que permita la circulación de intensidad, habiéndose de comprobar esto visualmente, si se ilumina el led LD1 de la placa, así como usando un multímetro en modo de corriente en serie con dos cables conectados en la *protoboard*. A continuación, se ha de volver a pulsar el interruptor y cotejar que el valor de intensidad es nulo y el led se ha apagado.

5.3 Prueba de medición de monóxido de carbono.

Puesto que la medición del monóxido de carbono es la base del dispositivo, es necesario comprobar que el sensor lo detecta correctamente y genera las señales adecuadas. Para ello, se utilizarán las funciones de medición de CO, temperatura y voltaje de referencia ya escritas y se mandará el valor medido al Debug viewer escribiéndolo como `printf`.

En ausencia de CO el valor deberá ser cercano al cero. Luego, se deben probar distintos valores de gas acercando una llama a una distancia prudencial del sensor, asegurándose de que el valor varía.

5.4 Prueba de conexión wifi

Mediante el Módulo USB-TTL CH340G se deberá verificar que el módulo ESP01s recibe comandos AT por el pin de recepción y envía una respuesta por el de transmisión, además de poderse conectar correctamente a una red wifi. Para ello, se deberá cambiar temporalmente la localización del cable conectado al pin TXD del módulo USB-TTL, colocándolo en la pista con el cable conectado al pin RX del módulo wifi. Se repetirá de forma análoga para el pin RXD de recepción. Finalmente, se conecta el conector USB tipo A del módulo a un puerto del mismo tipo de un pc.

Para enviar comandos y leer respuestas, se abrirá un terminal Tera Term en dicho pc y se seleccionará el puerto en el que esté conectado el USB serie CH340G. Se ajustará la tasa de baudios a 115200 bits/s y se agrupará la información en 8 bits, usando bit de parada pero no de paridad. Se debe escribir en primer lugar por teclado "AT" y pulsar *enter*, enviando este

comando al chip y recibiendo, si la comunicación es correcta, el mensaje "OK". A continuación, se enviará el comando "AT+CWMODE_CUR=1" para configurar el chip en modo estación y el comando "AT+CWLAP" para listar las IP al alcance. Si aparece una ristra de direcciones, significa que las funcionalidades wifi del chip funcionan. Por último, se debe escribir "AT+CWJAP=[<ssid>],[<pwd>]", con *ssid* el nombre de una red a la que se tenga acceso y *pwd* su contraseña. Si el chip responde "WIFI CONNECTED / WIFI GOT IP / OK" significará que las funcionalidades wifi funcionan.

5.5 Prueba giro del motor

Para cerciorarse de que el motor no presenta ningún defecto en su interior, se debe realizar una prueba de giro simple. Para ello, con el interruptor de la fuente en posición de corte, se usará un cable para enlazar el bus de masa de la placa de prototipado con el polo negativo del motor y otro para unir el bus de 5 V con el positivo. Acto seguido, se pulsará el botón para dejar circular la corriente y el motor deberá comenzar a girar a su velocidad máxima. Se debe dejar el motor en este estado por un tiempo no inferior a 30 segundos para determinar que no existe ningún problema. Por último, se debe repetir el proceso conectando los cables a los polos opuestos del motor para comprobar visualmente que gira en el sentido contrario.

5.6 Prueba conexión a servidor

Es recomendable comprobar que el servidor abierto en la aplicación QT envía y recibe correctamente datos. Para ello, se precisa únicamente de un pc con el programa desarrollado para el proyecto instalado.

El servidor envía y recibe comandos por protocolo TCP. Una vez iniciada la aplicación desde QT, se clicca en el botón de iniciar la comunicación por el puerto seleccionado. A continuación, se abre el Símbolo del sistema (*Command Prompt*) de Windows y se ejecuta el comando "Telnet localhost 1024" para conectarse al mencionado puerto de comunicación, (1024 en este caso). El servidor enviará un mensaje para indicar que se está intercambiando información. Seguidamente, en el programa del proyecto, se utiliza el ratón para acceder a la segunda pestaña, que contiene las lecturas de monóxido de carbono del sensor, y se busca el primer cuadro de texto, que normalmente muestra la medida más reciente de este gas. Ahora, se deben escribir valores numéricos por teclado en la aplicación intérprete de comandos y comprobar que se muestran en él.

6 PRUEBA GENERAL

Tras finalizar el montaje, se deben realizar las pruebas necesarias para comprobar el funcionamiento del dispositivo. Para llevarlas a cabo, se precisa, además de los elementos que componen el proyecto ya montados y cableados, del uso de algunas herramientas más. Estas son:

- Un PC con conexión a internet y el programa QT instalado.
- Un cable mini-usb a usb.
- Bomba de colección de gas.
- Una aplicación para emular un terminal (Tera Term, PuTTY).

Se deben realizar las pruebas en una estancia exenta de monóxido de carbono, abierta, para evitar acumulación de este gas, y a temperatura ambiente.

En primer lugar, se subirá al microcontrolador el código desarrollado empleando para ello el cable mini-usb a usb.

Para leer los mensajes que se envían desde el microcontrolador, se conectará el USB del módulo USB-TTL a un puerto del mismo tipo de un PC.

A continuación, se abrirá en Tera Term un terminal para comprobar en tiempo real los mensajes que se están enviando desde los pines UART. Se debe ajustar sus parámetros para que coincidan con los del chip ESP8266, entre ellos el *baud rate*, que ha de ser de 115200 bits/s, el envío de información en grupos de 8 bits, uso de bit de parada y sin bit de paridad. En cuanto al *setup*, la recepción de datos se dejará en automático y la transmisión en *Carriage Return + Line Feed* (CR + LF), es decir, moviendo el cursor al inicio de una línea siguiente.

Antes de encender la placa se ha de abrir el servidor desde la aplicación de QT o desde el ejecutable que se puede crear desde ella, puesto que el microcontrolador envía los comandos de conexión cada vez que se inicia. También es necesario que la señal wifi a la que se conectará el módulo (escrita en el programa) esté activa.

Se debe iniciar el programa desde la aplicación QT y pulsar el botón 'Abrir' para activar el servidor en el puerto. Luego se debe pulsar el interruptor del módulo de fuente de alimentación para permitir el paso de corriente y, a continuación, el botón *reset* negro de la placa. Se debe esperar a que el programa se ejecute y observar los valores medidos en la pantalla 'Medidas'.

Para probar distintos niveles de monóxido de carbono se debe emplear la bomba de colección de gas: se debe acercar su boquilla de cerámica a una llama y luego comprimir la

bomba para acumular el CO en su interior. Luego, se debe aproximarla al sensor y expulsar el gas.

Tras hacer esto se deben comprar varias cosas: que el valor de CO que se muestra por pantalla aumenta, que se guarda correctamente en el historial y que se puede exportar dicho historial a un fichero de texto, que el motor comienza a girar y que su valor de giro ya no es nulo.

4. PRESUPUESTO

Índice Presupuesto

1	TABLA DE PRECIOS ELEMENTALES	171
2	CUADRO DE PRECIOS DESCOMPUESTOS	171
3	ESTADO DE MEDICIONES	173
4	RESUMEN DEL PRESUPUESTO.....	173

1 TABLA DE PRECIOS ELEMENTALES

Tabla resumen con los precios unitarios sin IVA de los elementos utilizados en el desarrollo del proyecto.

1. Tabla de precios elementales			
Ref.	Unidad	Descripción	Precio (€)
Materiales			
m1	ud.	Placa de desarrollo STM32L053R8 + Placa de expansión P-NUCLEO-IKA02A1 (pack)	40,74
m3	ud.	Placa de expansión X-NUCLEO-IHM13A1	16,23
m4	ud.	Módulo wifi ESP-01S	3,75
m5	ud.	Motor DC Nidec serie DMN29, 12 V	23,55
m6	ud.	UART-TTL USB Adaptador CH340G	3,94
m7	ud.	Adaptador convertidor DC 9V	6,31
m8	ud.	Protoboard	2,63
m9	ud.	Modulo para fuente de alimentación de 3.3 V – 5 V	3,33
m10	ud.	Cables Dupont hembra/hembra	0,08
m11	ud.	Cables Dupont macho/hembra	0,08
m12	ud.	Cables jumper macho/macho	0,075
m13	ud.	Cable USB 2.0 a mini USB	2,76
m14	ud.	Codificador giratorio AMT103-V	21,58
Software			
s1	días	Keil µVision (MDK-Community edition)	0
s2	días	Qt for Application Development (316 € / mes)	10,53
s3	días	Tera Term	0
Mano de obra			
h1	h	Ingeniero electrónico	18,00

2 CUADRO DE PRECIOS DESCOMPUESTOS

Ref.	Unidad	Descripción	Precio (€)	Cantidad	Total (€)
d1	ud.	Montaje del conjunto de control, monitorización del monóxido de carbono y conectividad wifi y pruebas de funcionamiento.			
Materiales					
m1	ud.	Placa de desarrollo STM32L053R8 + Placa de expansión P-NUCLEO-IKA02A1 (pack)	40,74	1	40,74
m3	ud.	Placa de expansión X-NUCLEO-IHM13A1	16,23	1	16,23
m4	ud.	Módulo wifi ESP-01S	3,75	1	3,75
m7	ud.	Adaptador convertidor DC 9V	6,31	1	6,31
m8	ud.	Protoboard	2,63	1	2,63
m9	ud.	Modulo para fuente de alimentación de 3.3 V – 5 V	3,33	1	3,33
m10	ud.	Cables Dupont hembra/hembra	0,08	2	0,16
m11	ud.	Cables Dupont macho/hembra	0,08	9	0,72
m12	ud.	Cables jumper macho/macho	0,075	5	0,375
s3	días	Tera Term	0	1	0
Mano de obra					
h1	h	Ingeniero electrónico	18,00	8	144,00
Medios auxiliares					
% M.A. sobre costes directos			5%	218,25	10,91
Precio ejecución material (€)					229,16

Ref.	Unidad	Descripción	Precio (€)	Cantidad	Total (€)
d2	ud.	Montaje de la maqueta para la extracción de aire y pruebas de funcionamiento.			
Materiales					
m5	ud.	Motor DC Nidec serie DMN29, 12 V	23,55	1	23,55
m10	ud.	Cables Dupont hembra/hembra	0,08	1	0,08
m11	ud.	Cables Dupont macho/hembra	0,08	2	0,16
m12	ud.	Cables jumper macho/macho	0,075	4	0,3
m13	ud.	Codificador giratorio AMT103-V	21,58	1	21,58
s1	días	Keil µVision (MDK-Community edition)	0	0,5	0
Medios auxiliares					
% M.A. sobre costes directos			5%	117,67	5,88
Precio ejecución material (€)					123,55

Ref.	Unidad	Descripción	Precio (€)	Cantidad	Total (€)
d3	ud.	Montaje y conexionado del dispositivo completo.			
Materiales					
d1	ud.	Montaje del conjunto de control, monitorización del monóxido de carbono y conectividad wifi y pruebas de funcionamiento.	229,16	1	229,16
d2	ud.	Montaje de la maqueta para la extracción de aire y pruebas de funcionamiento.	123,55	1	123,55
Mano de obra					
h1	h	Ingeniero electrónico	18,00	0,25	4,5
Medios auxiliares					
% M.A. sobre costes directos			5%	357,21	17,86
Precio ejecución material (€)					375,07

Ref.	Unidad	Descripción	Precio (€)	Cantidad	Total (€)
d4	ud.	Desarrollo del código completo del microcontrolador y pruebas.			
Materiales					
s1	días	Keil µVision (MDK-Community edition)	0	24	0
s3	días	Tera Term	0	24	0
m6	ud.	UART-TTL USB Adaptador CH340G	3,94	1	3,94
m12	ud.	Cable USB 2.0 a mini USB	2,76	1	2,76
Mano de obra					
h1	h	Ingeniero electrónico	18,00	192	3456,00
Medios auxiliares					
% M.A. sobre costes directos			5%	3462,70	173,14
Precio ejecución material (€)					3635,84

Ref.	Unidad	Descripción	Precio (€)	Cantidad	Total (€)
d5	ud.	Desarrollo de la interfaz gráfica y pruebas.			
Materiales					
s2	días	Qt for Application Development (316 € / mes)	10,53	12	126,36
Mano de obra					
h1	h	Ingeniero electrónico	18,00	96	1728,00
Medios auxiliares					
% M.A. sobre costes directos			5%	1854,36	92,72
Precio ejecución material (€)					1947,08

3 ESTADO DE MEDICIONES

3. Estado de mediciones			
Ref.	Unidad	Descripción de la partida	Cantidad
d3	ud.	Montaje y conexionado del dispositivo completo	1
d4	ud.	Desarrollo del código completo del microcontrolador y pruebas	1
d5	ud.	Desarrollo de la interfaz gráfica y pruebas	1

4 RESUMEN DEL PRESUPUESTO

4. Valoración total					
Ref.	Unidad	Descripción	Precio (€)	Cantidad	Total (€)
d3	ud.	Montaje y conexionado del dispositivo completo	375,07	1	375,07
d4	ud.	Desarrollo del código completo del microcontrolador y pruebas	3635,84	1	3635,84
d5	ud.	Desarrollo de la interfaz gráfica y pruebas	1947,08	1	1947,08
Precio ejecución material (€)					5957,99
Beneficio industrial			13%		774,54
Gastos generales			6%		357,48
Total presupuesto sin IVA (€)					7090,01
Impuesto sobre el Valor Añadido (IVA)			21%		1488,90
Total presupuesto con IVA (€)					8578,91

El presupuesto de ejecución material de este proyecto asciende a la cantidad de

Ocho mil quinientos setenta y ocho euros con noventa y un céntimos (8578,91€).

