



**Universidad
de La Laguna**

ESCUELA SUPERIOR DE INGENIERÍA Y TECNOLOGÍA

Trabajo de Fin de Grado

DISEÑO DE DISPOSITIVO DE BAJO CONSUMO PARA LA
LECTURA DE GASES DENTRO DE UNA GALERÍA

Titulación: Grado en Ingeniería Electrónica Industrial y Automática

Autores: Ezequiel Cortés Martín

Alberto Doro García

Tutores: Jose Francisco Gómez González

Santiago Torres Álvarez

Prefacio

Este proyecto nos ha servido para aprender a aprender. Hemos comprendido que el papel lo sostiene todo, pero la práctica es mucho más cruda. No siempre todo sale a la primera, de hecho, la gran mayoría de veces hay que trabajar a modo prueba y error hasta conseguir el resultado final ansiado. Además, entendimos que no todo se reduce a la formación académica, pues hay numerosos factores que juegan un papel vital a la hora de realizar un diseño, lo que puede extrapolarse, incluso, a otras facetas de la vida. La experiencia, la comunicación, la gestión del estrés, el presupuesto, a veces ajustado, y las circunstancias intrínsecas del entorno son fundamentales para el desarrollo de cualquier trabajo. Pero, sobre todo, hemos entendido a la fuerza que el factor humano y la organización son los papeles principales para que un proyecto salga bien. Es imprescindible conocer las inquietudes, problemas, fortalezas y debilidades de los compañeros de trabajo, saber ser empático y ponerse de verdad en la piel de la otra persona para comprenderla y darle apoyo, son elementos esenciales a fin de superar los objetivos marcados. De hecho, nos atreveríamos a decir que, sin jugar estas bazas correctamente, desarrollar y finalizar con éxito un proyecto como este es casi imposible. Por esta razón, aparte de que esta experiencia nos ha servido para reflejar los conocimientos adquiridos en estos cuatro años, sobre todo nos ha permitido quitarnos la venda de lo teórico y acercarnos a la realidad profesional, haciéndonos más capaces y humanos.

Contenido

1. Abstract.....	4
2. Introducción	4
3. Objeto del proyecto.....	5
4. Peticionario.....	6
5. Requerimientos del peticionario	6
6. Antecedentes.....	6
7. Normativa aplicable.....	8
8. Estudios previos	8
9. Diseño definitivo.....	11
9.1. Componentes del dispositivo	11
9.2. Arduino	20
9.3. Protocolos de comunicación	21
9.5. Validación de datos	29
9.6. Esquema eléctrico general	36
9.7. Consumo del dispositivo	38
9.8. Diseño del circuito impreso	38
9.8.1. Programa de diseño	38
9.8.2. Esquemático y diseño de la PCB	39
10. Encapsulado	41
11. Presupuesto.....	43
11.1. Introducción.....	43
11.2. Costes materiales	43
11.3. Costes de mano de obra	44
11.4. Coste de ejecución material	44
11.5. Gastos generales	45
11.6. Beneficios industriales.....	45
11.7. Impuestos	46
11.8. Presupuesto de contrata.....	46
12. Conclusiones	47
13. Bibliografía	47
Pliego de Condiciones.....	50
ANEXOS	58

1. Abstract

The project consists of the design of a measurement system, autonomous and low consumption to measure gases, temperature and humidity inside a gallery. The scope of the project has been directed to be able to know how the gases vary inside the galleries, to increase the security of employees and to reduce costs when opening the gallery.

Firstly, it has been established as minimum goals fulfil with the client requirements and overcome them.

Secondly, a component selection process has been carried out for the design of the device, making a study of the market and the needs of the device.

Thirdly, a cyclical system has been chosen which consists of reading the measurements from the sensors, processing the data and adapting it to information of interest to the user and sending it to a removable memory card in easy to work files. Then, the device goes into power down mode until the new measurement.

Finally, the device's PCB has been designed and a way of encapsulating it has been proposed to ensure the protection of the measurement system.

2. Introducción

Las galerías filtrantes o de captación de agua subterránea, a diferencia de los pozos que son verticales, son excavaciones subhorizontales en forma de túnel, con ligera pendiente decreciente hacia la bocamina. Se puede llegar a profundizar varios kilómetros hasta encontrar agua.

El caudal que se obtiene de estas obras subterráneas es variable en el tiempo. Al principio, suele ser elevado, hasta que llega el momento en que se estabiliza y se comporta en función de la carga de agua que recibe. En ocasiones, la demanda de agua es mayor de la que puede ofrecer la galería, por lo que se suele seguir perforando hasta llegar al equilibrio de oferta – demanda. Sin embargo, las galerías no son fuentes de agua inagotables, llegando a veces a la coyuntura de secarse.

En el Archipiélago Canario existen más de 1500 galerías con una longitud mayor de 2000 kilómetros. En los años sesenta, dos tercios de los recursos hídricos consumidos en las Islas Canarias eran sustentados gracias a estas obras. No obstante, es cierto que en algunas islas se han agotado prestamente, como en Fuerteventura y Lanzarote. En otras como la isla de Tenerife (donde se localiza más del setenta por ciento del total de galerías), sigue siendo el método más utilizado como aprovechamiento del agua.

Sin embargo, aunque las excavaciones del subsuelo proporcionan la mayoría del agua necesaria, menos de la mitad están activas, reduciéndose cada vez más a medida que pasan los años. Aunque si se realiza una reflexión, se puede llegar a la conclusión de que la disminución en la captación de aguas subterráneas asiste indirectamente al equilibrio entre las extracciones y recargas. Pero la realidad es que a medida que las galerías crecen, se encarecen las labores de perforación. Además, cada vez es más caro acondicionar el lugar de trabajo, teniendo que ser consecuentes y debiendo de aplicar todas las medidas preventivas necesarias, para disminuir el riesgo de peligro de los trabajadores lo máximo posible. Esto propicia a que muchos dueños de galerías no puedan abrirlas, dejando así muchas galerías inactivas.

3. Objeto del proyecto

El objetivo principal del proyecto es el diseño de un sistema de recogida de datos, de la concentración de gases en una galería de agua, a varios cientos de metros o kilómetros de la bocamina.

Para alcanzar este objetivo principal hay varios objetivos secundarios:

- Selección de los sensores que permitan medir monóxido de carbono (CO), dióxido de carbono (CO₂), oxígeno (O₂), humedad, temperatura y presión en las condiciones extremas del interior de una galería de agua, de forma ininterrumpida por periodos largos de tiempo (varios meses).
- Diseño del sistema de control y adquisición de datos de los sensores durante periodos largos de tiempo.
- Diseño del protocolo y método de almacenamiento de la información en algún sistema extraíble.
- Estudio de la demanda energética del sistema y diseñar la fuente de energía con autonomía suficiente.

- Estudio de la posibilidad de extracción de energía eléctrica del movimiento del agua de la galería que va del naciente hacia la bocamina.

4. Peticionario

El petionario del presente proyecto es el Departamento de Ingeniería Industrial de la Universidad de La Laguna, situado en la avenida Astrofísico Francisco Sánchez, del municipio de San Cristóbal de La Laguna, en Santa Cruz de Tenerife.

5. Requerimientos del petionario

El petionario solicita un mecanismo que sea capaz de funcionar durante largos periodos de tiempo dentro de una galería, con el propósito de recoger información de las concentraciones de los gases de CO₂, CO y O₂ y temperatura y humedad. Como requisito indispensable, la toma de datos deberá de cumplir las siguientes características:

<i>Tabla 1: Requerimientos</i>		
GAS	RANGO (ppm)	RESOLUCIÓN (ppm)
O ₂	0 – 21000	≤ 20
O ₂	0 – 4000	≤ 5
CO ₂	0 – 1000000	≤ 100

6. Antecedentes

Las actividades que se desempeñan dentro de las galerías, desde su construcción hasta su mantenimiento, se realizan en unas condiciones desfavorables, llegando a ser peligrosas e incluso mortales. La cantidad de peligros que se pueden desencadenar dentro de una galería son múltiples, considerándose espacios confinados y teniendo que actuar en función de los protocolos establecidos. No solo existe el riesgo obvio del desplome de la galería, que es uno de los más comunes debido a las peculiaridades propias de estos espacios, pues son túneles

grabados en la piedra. Además de los riesgos físicos debidos a las características intrínsecas de estos espacios, tales como golpes, caídas, fatiga física o incluso electrocución; también existen riesgos de intoxicación, ya sea por concentraciones de sustancias tóxicas por encima del límite de seguridad, atmósferas irritantes y corrosivas, etcétera.

Existen medidas preventivas que amortiguan el riesgo de estos peligros, pero en ocasiones pueden llegar a ser precarias, sobre todo en los riesgos por intoxicación. Entre estas, se encuentra comprobar la concentración de oxígeno y verificar la ausencia de posibles gases tóxicos, tales como el CO₂ o el CO, u otras medidas relevantes como la presión, temperatura y humedad. Para ello, se utilizan medidores comerciales a tiempo real como Equipos de Protección Individual y se realiza ventilación forzada. El principal problema de estas medidas es que, con un dosímetro, solo puedes conocer la concentración de gases en el aire a medida que vas profundizando en la galería. Asimismo, se genera un estado de incertidumbre, debido a la complejidad que conlleva estimar las medidas necesarias de caudal y tiempo de ventilación, para que el acceso a una galería sea seguro. De hecho, en la mayoría de las ocasiones, se hacen aproximaciones a tanteo, confiando en la experiencia propia de los trabajadores, sin saber a ciencia cierta la seguridad que existe en el momento de entrar a estos lugares. Además, las concentraciones de los gases no son constantes a lo largo del año, si no que dependen de muchos factores que hacen que vayan cambiando.

Esto supone una gran inversión de capital cada vez que se quiera abrir la galería para realizar cualquier trabajo de mantenimiento, voladuras, excavaciones, etc. Esto se produce, en parte, a no poder calcular la cantidad de aire a insuflar, debiendo tener los equipos encendidos por más tiempo o con más potencia de la necesaria, suponiendo el deterioro de ellos, el combustible gastado y las horas perdidas de mano de obra.

Ante esta dificultad, resulta práctico y lógico tener la posibilidad de recoger información sobre cómo cambian las concentraciones de los gases en la galería, a través de su recorrido en función del tiempo. De esta manera, se podría realizar un estudio para conocer en qué periodo del año es más conveniente abrir la galería. Así, se incrementa la seguridad del trabajador y, al mismo tiempo, se puede calcular el caudal y tiempo necesario para ventilar la galería hasta unas condiciones óptimas para el trabajo. Esto implica una reducción de costes, tanto en carburantes necesarios para las bombas de ventilación como en horas de mano de obra.

7. Normativa aplicable

Las normativas aplicables a este proyecto [1]:

- Componentes electrónicos – Símbolos IEC 60617
- Componentes electrónicos – Valores normalizados IEC 60063 UNE 20531
- Componentes electrónicos – Marcado IEC 60062 UNE-EN 60062
- Componentes electrónicos – Marcado Circuitos Integrados
- Componentes electrónicos – Encapsulado JEDEC - JESD30-B
- Diseño de circuito impreso. IPC 221
- Fabricación de circuito impreso. IPC 4101
- Compatibilidad electromagnética, norma IEC 61000 – UNE-EN 61000.
- Sostenibilidad medioambiental -RoHS
- Sostenibilidad medioambiental -RAEE

8. Estudios previos

Los equipos existentes en la actualidad para la medida de concentraciones de gases, autónomos y apropiados para el agresivo entorno de una galería son muy escasos. A este hecho se suma que, en la mayoría de las ocasiones, deben obtenerse bajo pedido y a medida, lo que supone un sobre coste y merma su rentabilidad.

Por esta razón, se ha propuesto el diseño de un dispositivo que se adapte a las necesidades de trabajar dentro de este tipo de espacios confinados. Por lo tanto, deberá de soportar temperatura elevadas (en torno a 40°C) y porcentajes de humedad altos. A su vez, se requiere que pueda operar por largos periodos de tiempo sin interrupciones, ya que la apertura continuada de la galería para recoger las lecturas de los gases sería inviable.

En la primera etapa del proyecto, se ha realizado un estudio previo del mercado para tratar de dimensionar el dispositivo de una manera adecuada y justificar su elección. Posteriormente, en el apartado 9. *Diseño definitivo*, se detallarán los componentes seleccionados.

- Sensores:

Antes de explicar qué es un sensor, resulta indispensable entender antes qué es un transductor. Un transductor se define como aquel dispositivo que es capaz de convertir una variable física en otra que tiene un dominio diferente. A este fenómeno se le denomina principio de transducción. Por lo tanto, decimos que un sensor es un dispositivo cuyo funcionamiento se basa en dicho principio, capacitado para detectar estímulos externos o acciones y responder en consecuencia, dando una salida útil para ser usada como variable de entrada a un sistema de procesamiento de la información. Es decir, se encarga de medir una magnitud física o química y transformarla, generalmente, en señales eléctricas. Así, por ejemplo, un microcontrolador puede interpretar dichas señales y acondicionarlas para que podamos captar información del medio que nos rodea de forma legible.

En primer lugar, se seleccionaron los sensores de forma que se adaptaran a las necesidades de la galería y que cumplieran con las especificaciones técnicas del cliente. Una vez fueran elegidos, se podría condicionar el resto de los componentes en base a los sensores, y no al revés. A fin de procurar reducir costes, se barajó la posibilidad de calibrar los sensores particularmente. Sin embargo, no fue posible. Pues los laboratorios no contaban con las exigencias que se solicitaban, y comprar un instrumento calibrado para calibrar otro dejaba de cobrar sentido en la reducción de costes. En consecuencia, se optó por buscar los sensores ya calibrados. Fue una tarea complicada, pues ninguna empresa nacional los comercializaba. De hecho, se habló con algunas compañías que trabajaban expuestos a gases tóxicos, pero todas utilizan equipos comercializados que no se adaptaban a las necesidades pertinentes. Se extendió la búsqueda de los sensores a países extranjeros. Finalmente, se encontró una única empresa que comercializaba los sensores ya calibrados y sin estar montado previamente en un equipo de medida comercializado.

- Microcontrolador:

Un microcontrolador (MCU en adelante) es un dispositivo programable. Computadoras en un solo chip que poseen una Unidad Central de Procesamiento [2] (CPU, *Central processing unit, en inglés*), memoria de programa y de datos, puertos de entrada/salida (E/S) serie y paralelo, temporizadores e interrupciones externas e internas.

La CPU es el cerebro (microprocesador) del microcontrolador. Se ocupa en términos generales de traer e interpretar las instrucciones de un programa informático para hacer que se

ejecuten. También incorpora una (o varias) Unidad Aritmética y Lógica (ALU, *Arithmetic and Logic Unit*), para realizar operaciones lógicas y aritméticas.

El programa que es escrito por el usuario o programador se almacena en la memoria de programa. Esta memoria es de tipo no volátil, es decir, los datos permanecen aunque desconectes la energía. Las memorias de programa más utilizadas actualmente en los microcontroladores son la EEPROM (*Electrically Erasable Programmable Read Only Memory*), y la EEPROM Flash.

Los datos temporales usados por los programas se guardan en la memoria de datos. A diferencia de la memoria de programa, esta sí es normalmente volátil, o sea, los datos se pierden al desconectar la alimentación. La memoria de datos se denomina Memoria de Acceso Aleatorio (*RAM, Random Access Memory*). Es una memoria de propósito general y se usa, generalmente, para almacenar los datos usados por los programas.

Las interrupciones permiten que el MCU responda a eventos internos y externos. Cuando ocurre una interrupción, el MCU sale del funcionamiento típico del programa para saltar y ejecutar el programa que hay establecido en el ISR o rutina de atención a la interrupción (*Interrupt Service Routine*, en inglés). Una vez finalizada la ejecución del ISR, el programa reanuda su flujo normal de ejecución.

Una vez escogidos los sensores, se decidió elegir el microcontrolador. Como requisito mínimo, era indispensable que, como mínimo, debía poseer tres seriales programables USART (*Universal Synchronous/Asynchronous Receiver Transmitter*) y un maestro/esclavo SPI (*Serial Peripheral Interface*). Estos protocolos de comunicación son explicados con más detalle en el apartado *9.3 Protocolos de Comunicación*.

En una primera instancia, se preseleccionó un PIC32MZ de la casa Microchip. Pues cumplía con los puertos E/S necesarios para comunicarse con los sensores y, a su vez, con algún dispositivo de almacenamiento extraíble. Asimismo, el consumo de potencia del MCU cuando operaba normalmente era asumible, sobre los 9 mA condicionado a 10Mhz como señal de reloj.

Sin embargo, después de analizar más en profundidad la programación y el alcance del PIC32MZ y otros MCU, se llegó a la conclusión de que utilizar el ATMEGA - 328P era mejor opción. Aunque sus características generales son más precarias que las de la primera elección, y la potencia consumida es casi la misma, la diferencia de coste es apreciable. De hecho, si se

llega a comparar precios, se puede estimar que, por cada compra de un PIC32MZ, se podrían comprar alrededor de cuatros ATMEGA 328 - P. Además, el ATMEGA tiene la ventaja de ser más accesible que el PIC, facilitando su compra en tiendas locales. A la larga esto supondrá un ahorro económico considerable, pues la idea es la futura fabricación en masa del dispositivo diseñado para situarlo en diferentes recorridos de las galerías. Por ejemplo, si una galería tiene una longitud de 3 kilómetros y se desea poner un equipo a cada 200 metros, se deberá de colocar 15 dispositivos.

- Almacenamiento extraíble

El dispositivo de almacenamiento de las lecturas de los sensores debe de ser extraíble. Es decir, debe de utilizarse una memoria no volátil, para que cuando se deje de suministrar energía eléctrica se conserven los datos guardados. En este caso, se va a utilizar una tarjeta de memoria flash, concretamente, una tarjeta microSD.

- Hidrogeneradores

Para procurar aumentar el servicio de las baterías y, al mismo tiempo, incorporar al proyecto la obtención de energías limpias y renovables, se ha propuesto utilizar la perpetuidad de agua corriente para generar energía a través de hidrogeneradores.

Se realizaron pruebas para estimar el caudal de agua de la galería y así calcular el número de hidrogeneradores y la altitud de la caída del agua necesaria para la recarga de las baterías, en función de la sección de manguera que se fuera a utilizar. Sin embargo, debido a la inexactitud del caudal se decidió dejarlo fuera del alcance de este proyecto, pero sí se ha propuesto en el esquema general del equipo como añadido, invitando a otros colaboradores a que en un futuro próximo los incorporen.

9. Diseño definitivo

9.1. Componentes del dispositivo

- Microcontrolador ATMEGA 328 - P:

El cerebro del dispositivo es el ATMEGA 328-P, un chip creado por Atmel y perteneciente a la serie mega AVR. Su arquitectura es RISC (*Reduced Instruction Set Computing*) de 8 bits. Es un MCU versátil, que ejecuta instrucciones en un solo ciclo de reloj,

llegando a alcanzar rendimientos cercanos a 1 MIPS por MHz, equilibrando el consumo de energía con la velocidad de procesamiento.

<i>Tabla 2: Especificaciones del MCU</i>	
Nombre	Valor
Program Memory Type	Flash
Program Memory Size (KB)	32
CPU Speed (MIPS/DMIPS)	20
SRAM (B)	2048
Data EEPROM/HEF (bytes)	1024
Digital Communication Peripherals	1-UART, 2-SPI, 1-I2C
Timers	2 x 8-bit, 1 x 16-bit
Capture/Compare/PWM Peripherals	1 Input Capture, 1 CCP, 6PWM
Number of Comparators	1
Temperature Range (°C)	-40 a 85
Operating Voltage Range (V)	1.8 a 5.5
Pin Count	32
Low Power	Yes

- **Sensores:**

Los sensores seleccionados han sido comprados a la empresa *CO2METER*. Cuentan con un certificado de calibración que garantizan la correcta lectura de los datos y tiene una garantía de 90 días. En total, han sido tres sensores, uno para medir cada concentración de gas

(CO₂, CO, O₂). No ha sido necesario comprar sensores de temperatura y humedad porque los sensores seleccionados ya vienen equipados para tomar medidas de temperatura, humedad, fecha y hora como añadido.

Los sensores son:

- Sensor ExplorIR®-M 100% CO₂

<i>Tabla 3: Especificaciones sensor CO2</i>	
Especificaciones	
Rango	0 - 1.000.000 ppm
Precisión	+/- 300 ppm / +/-5%
Resolución	100 ppm
No linealidad	< 1%
Tiempo calentamiento	1.2 seg
Condiciones de funcionamiento	(0-50) °C (0-95) %HR
Tensión de trabajo	(3.25 - 5) V
Consumo	33ma pico, <3.5mW promedio



CM-40830

Figura 1: Sensor de CO2

- Sensor de EC inteligente Alphasense con 25% de oxígeno

<i>Tabla 4: Especificaciones sensor O2</i>	
Especificaciones	
Rango	0 - 20.9%
Precisión	+1%
Resolución	10 ppm
No linealidad	< 0.6%
Tiempo calentamiento	10 seg
Condiciones de funcionamiento	(-30 / 60) °C (0-99) %HR

Tensión de trabajo	(3.3 -5.5) V
Consumo	10ma pico, <3mW promedio



CM-32910

Figura 2: Sensor de O₂

- Alphasense 5,000ppm Sensor inteligente de monóxido de carbono EC:

<i>Tabla 3: Especificaciones sensor CO</i>	
Especificaciones	
Rango	0-5000ppm
Precisión	+/-1%
Resolución	1 ppm
No linealidad	<2.5%

Tiempo calentamiento	10 seg
Condiciones de funcionamiento	(-30 / 60) °C (0-99) %HR
Tensión de trabajo	(3.3 -5.5) V
Consumo	10ma pico, <3mW promedio



CM-31910

Figura 3: Sensor de CO

Todos estos sensores tienen certificado de calidad ISO 9001 por lo que sus medidas vienen ya acondicionadas y no será necesario implementar más electrónica sobre ellos para obtener sus medias.

- Transistor de efecto de campo metal-óxido-semiconductor [3] (MOSFET):

Los transistores de efecto campo (FET) son dispositivos que se pueden utilizar como amplificadores o interruptores lógicos. En este proyecto, se van a utilizar transistores MOSFET de canal n (NMOS) como interruptores lógicos. Los terminales de este transistor son el drenador, la puerta, la fuente y el sustrato. Sin embargo, en el funcionamiento normal de estos dispositivos, la corriente que pasa por el terminal del sustrato es despreciable, pudiendo considerarse que está conectado a la fuente, teniendo así un dispositivo de tres terminales.

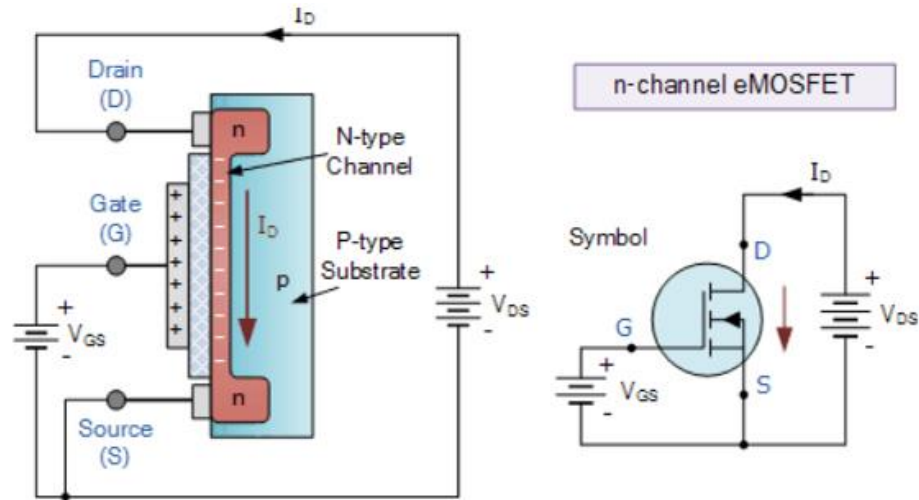


Figura 4: "Esquema transistores MOSFET"

La potencia disipada por el MOSFET se aplica directamente con la resistencia entre el drenador y la fuente y la corriente al cuadrado que lo atraviesa: $P = R_{DS} * I_D^2$

Las regiones de funcionamiento son: región de corte, región óhmica y región de saturación. En nuestro caso, queremos que estos transistores permutan entre la región de corte y la región de saturación a nuestra conveniencia.

- Región de corte:

El transistor permanecerá en la región de corte mientras la unión drenador - sustrato esté polarizada en inversa por el generador V_{DS} . Cuando la tensión que se aplica a la puerta (G) en función de la fuente (S), V_{GS} , supere la tensión de umbral V_{TO} , dejará de estar en dicha región.

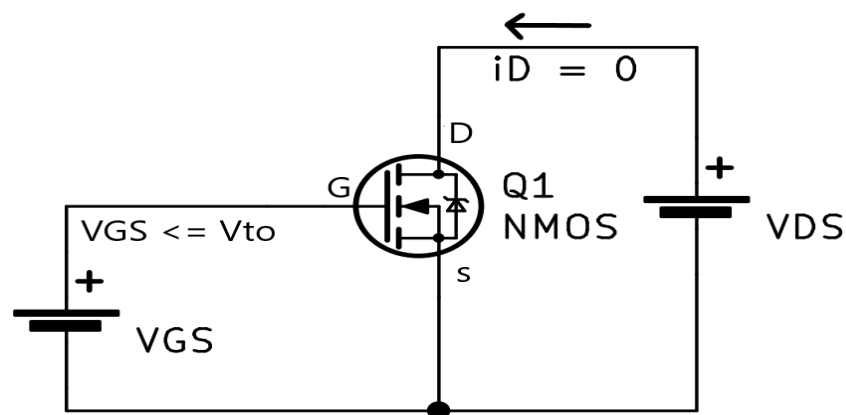


Figura 5: "Conexión del MOSFET para operar en región de corte"

Por lo tanto, cuando está en la región de corte, funciona como un interruptor abierto entre la fuente y el drenador. De hecho, la potencia que disipa es cero, pues, aunque la resistencia entre el drenador y la fuente dependen de V_{GS} , la corriente I_D es cero.

- Región óhmica:

Cuando V_{GS} supera la tensión de umbral V_{TO} , se forma un canal de tipo n entre el drenador y la fuente. Luego a medida que se aumenta V_{DS} , la corriente pasa por el drenador a través de la fuente y el canal. El NMOS se comporta como una resistencia variable conectada entre el drenador y la fuente, que disminuye a medida que aumenta V_{GS} o viceversa.

En esta región, la potencia disipada es la más grande que en las otras dos regiones, pues la corriente I_D puede ser grande y la resistencia entre el drenador y la fuente no tiene por qué ser pequeña. Por lo tanto, hay que evitar entrar en esta región de funcionamiento para utilizar el transistor como interruptor lógico.

- Región de saturación:

A medida que se aumenta la tensión V_{DS} , la tensión V_{GD} disminuye. Cuando V_{GD} se hace igual a la tensión de umbral V_{TO} , se entra en la región de saturación, pues la anchura del canal en el extremo del drenador se hace cero, por lo que la resistencia entre drenador y fuente se hace muy pequeña (idealmente cero). Luego una vez entra el transistor en saturación, independientemente del incremento de tensión de V_{DS} , la corriente I_D es constante. Es decir, se puede suponer que el MOSFET se comporta como un cortocircuito entre la fuente y el drenador. Entonces, aunque la corriente I_D pueda ser grande, la resistencia R_{DS} es demasiado pequeña, por lo que la potencia que disipa es muy baja también.

En el caso de este diseño, se van a utilizar transistores MOSFET de *Logic Level* tipo n, concretamente, los IRL540N, los cuales son capaces de trabajar entre corte y saturación con una señal de control TTL, perfectos para controlar el paso de la corriente con un pin del microcontrolador conectado a la puerta del transistor.

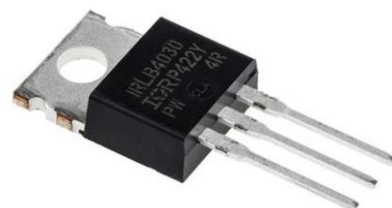


Figura 6: "MOSFET IRL 430"

- Tarjeta microSD:

La tarjeta microSD es una tarjeta de memoria flash cuyas medidas son 15×11×1 milímetros, con un área de 165 mm². La memoria flash permite velocidades de transmisión mayores que la memoria EEPROM, pues permite la lectura y escritura de múltiples posiciones de memoria en la misma operación. Es una memoria de tipo no volátil, por lo que se puede transportar la tarjeta con toda la información guardada a otro lugar sin perder los datos al desconectarlo de la fuente de alimentación. Además, en caso de falla del equipo, los datos no se perderán, por lo que da una alta confiabilidad.



Figura 7: "Tarjeta microSD"

- Batería:

Las baterías seleccionadas son baterías de ácido de plomo con dimensiones 151x50x95 milímetros, de 12Ah a 6 voltios. Se ha elegido esta tensión de referencia para reducir la diferencia de tensión lo máximo posible entre las baterías y el dispositivo de medida a diseñar, reduciendo así el gasto energético en los reguladores de tensión, pues a menor *dropout*, menos potencia se disipa por estos componentes. Por otro lado, elegir baterías con la menor tensión de referencia posible, facilita la posible carga de estas con la energía proveniente de los hidrogenadores.



Figura 8: "Baterías Panasonic de 12Ah"

Hay otros componentes que forman el aparato de medida, como componentes pasivos (resistencias, condensadores, cristal de cuarzo), componentes electromecánicos (pulsador para reiniciar el MCU), zócalos para la tarjeta de memoria microSD y el MCU y los reguladores de tensión.

9.2. **Arduino**

En este proyecto, se ha utilizado la placa de desarrollo Arduino UNO para la programación del microcontrolador [4]. Arduino es una plataforma de creación de electrónica de código abierto. Está basada tanto en hardware como software abierto, enfocada así a creadores y desarrolladores. Es una placa basada en un microcontrolador ATMEL y puede ser programada utilizando el entorno de programación Arduino IDE [5].

El entorno de programación Arduino IDE, es el usado por defecto para todas las placas de desarrollo Arduino. En él, se puede programar de una forma sencilla y rápida, a un nivel de abstracción muy elevado. A diferencia del lenguaje de programación de sus microcontroladores, que es un nivel muy bajo.

Estas dos características, son las que han llevado a las placas Arduino a ser líderes mundiales a la hora de realización de prototipos, gracias a su facilidad de uso.



Figura 9: “Microcontrolador Arduino UNO”

9.3. Protocolos de comunicación

Durante la ejecución del programa, el microcontrolador necesitará obtener información de otros componentes, como son los sensores, y a su vez, necesitará volcar información hacia otros componentes, como es la tarjeta microSD. Es una memoria de tipo no volátil, De esta manera, se puede transportar la tarjeta con toda la información guardada a otro lugar. Además, en caso de falla del equipo, los datos no se perderán, por lo que da una alta confiabilidad.

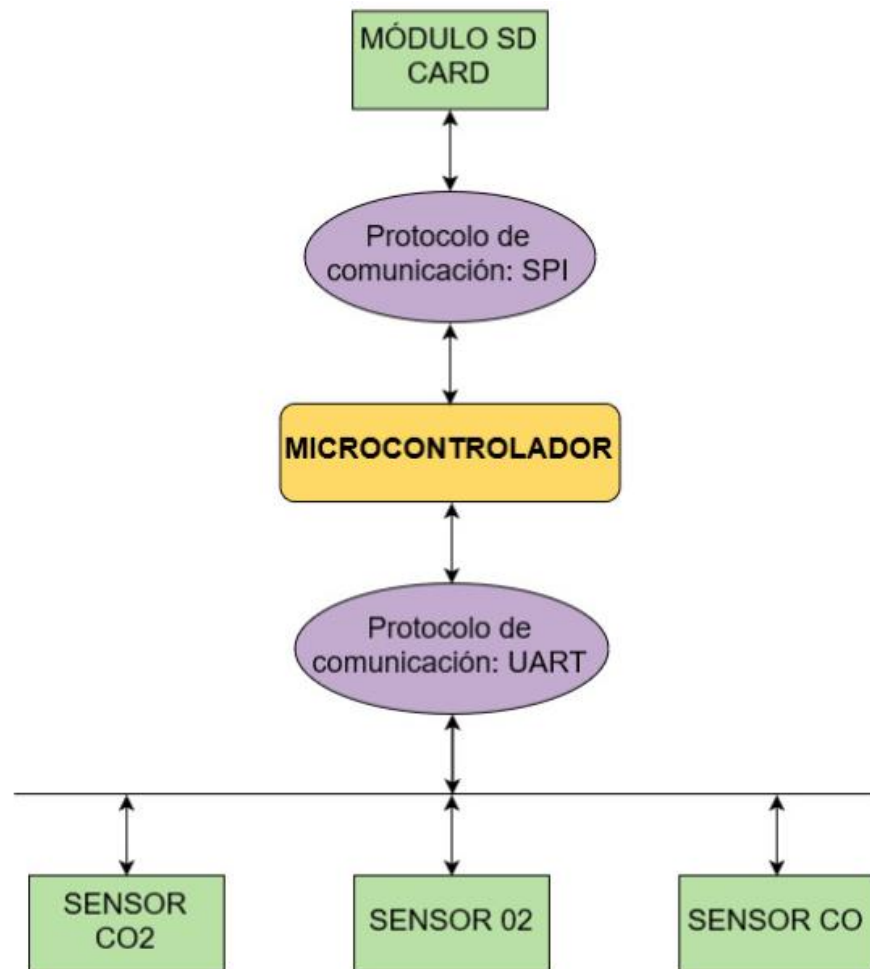


Figura 10: "Esquema general de protocolos de comunicación utilizados"

Para esto, es necesario que el microcontrolador sea capaz de gestionar los diferentes protocolos de comunicación.

Se conoce como protocolo de comunicación [6], cualquier sistema de normas que permita mediante ellas, comunicar a dos o más sistemas a través de un medio. Estas reglas suelen estar basadas en la transmisión de la información a través de cualquier tipo de variación

de una variable física. Dichas reglas tratan de definir la sintaxis, semántica y sincronización utilizadas durante una comunicación, para que, de esta manera, las entidades sean capaces de comunicarse. Además, se pueden definir métodos de detección de fallos, para detectar cuando no se ha realizado el envío de forma correcta.

Gracias a estos protocolos de comunicación, es posible comunicar equipos de forma sencilla entre ellos, tan solo utilizando el mismo protocolo de comunicación en todos los equipos de una red.

Con los componentes seleccionados para el diseño, los protocolos necesarios para la comunicación serán el protocolo UART (*Universal Asynchronous Receiver-Transmitter*), utilizado para la comunicación con los sensores y el protocolo SPI (*Serial Peripheral Interface*) usado para la comunicación con la tarjeta SD.

- Comunicación serie:

Ambos protocolos utilizados son comunicaciones seriales. Una comunicación serial es aquella en la que sus datos son enviados uno tras otro, para esto y para que las comunicaciones sean posibles, es necesario algún tipo de sincronización, para poder realizar la comunicación de una forma correcta, esta sincronización se suele llevar a cabo a través de relojes. Debido a la necesidad de sincronización, se desarrollaron dos tipos de comunicaciones seriales diferentes.

- Comunicación serial síncrona:

Esta variante de las comunicaciones seriales, se basa en que tanto el transmisor como el receptor comparten una misma señal de sincronismo. De esta manera, se tiene una fiabilidad muy alta de que las comunicaciones no tengan errores, ya que las señales de sincronismo de ambos dispositivos serían idénticas. Por otra parte, este tipo de sincronización entre dispositivos, al requerir la conexión de la misma señal de reloj a ambos dispositivos, no es eficaz para la transmisión a largas distancias.

- Comunicación serial asíncrona:

Esta segunda variante de las comunicaciones seriales, se basa en no tener ninguna fuente de reloj común entre los dispositivos. Ambos extremos de la comunicación tendrán sus propias señales de sincronismo, pero estas han de ser exactamente iguales en frecuencia para poder crear una comunicación. Gracias a esta característica de este tipo de comunicación serial, se

pueden enviar mensajes a mayores distancias, pero será necesario el ajuste de los relojes de ambos extremos, para que se pueda llevar a cabo la comunicación.

- Protocolo UART:

Este protocolo de Transmisor Receptor Asíncrono Universal permite comunicar dos dispositivos con tan solo dos cables de datos, Rx y Tx. Además, al tratarse de un protocolo asíncrono, como ya se explicó anteriormente, no es necesario el uso de una señal de sincronismo. Sin embargo, existe un término utilizado en este protocolo llamado tasa en baudios, que ayuda a que los dispositivos que se estén comunicando permanezcan sincronizados, al fijar una velocidad de transferencia de datos. Esta velocidad hace referencia al número de bits transmitidos por segundo. La principal limitación de este método de intercambios de datos es que solo dos dispositivos pueden estar conectados. Esto es así ya que, el pin de transmisión Tx de un dispositivo, se conectará al de recepción del otro y viceversa, quedando así todos los pines ocupados. Para realizar otra comunicación con otro dispositivo, serían necesarios otros 2 pines Rx y Tx en el dispositivo.

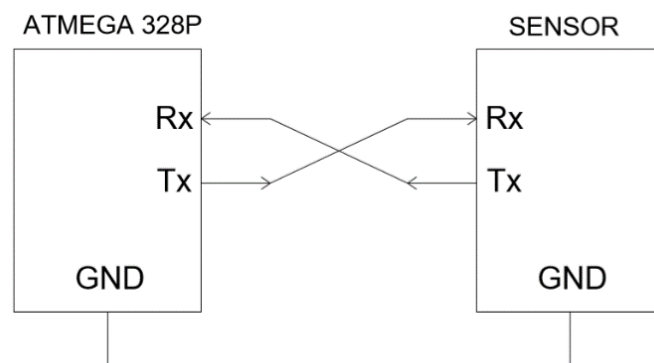


Figura 11: "Conexiones del protocolo UART".

La configuración establecida del puerto serie es la que viene por defecto, ya que se utilizó el entrenador Arduino, donde se puede programar en un nivel de abstracción superior. Luego la configuración es modo asíncrono 9600/8N1. Es decir, la tasa de transmisión es de 9600 bps, hay 8 bits de datos, no tiene bit de paridad y cuenta con un bit de parada. La frecuencia del oscilador es de 16 MHz.

- Protocolo SPI:

A diferencia del protocolo anteriormente mencionado, el *Serial Peripheral Interface* es un protocolo síncrono, por lo que es necesario que todos los dispositivos conectados a la red de comunicación estén conectados al mismo reloj. Para la implementación de este protocolo es necesario conocer las cuatro señales con las que trabaja. En primer lugar, como ya se ha comentado, al ser una comunicación síncrona es necesaria una señal compartida de reloj, esta señal es la encargada de definir la velocidad del bus, es decir, la velocidad de envío de cada bit también llamada SCK. En segundo lugar, están las dos señales de datos MISO (*Master In Slave Out*) y MOSI (*Master Out Slave In*). Al contrario que en las comunicaciones UART, los integrantes de esta comunicación no se encuentran en el mismo nivel de jerarquía, ya que este se basa en una topología Maestro/Esclavo. Por esto, son necesarias las líneas MISO (salida del maestro y entrada del esclavo), y MOSI (salida del esclavo y entrada del maestro). Por último, gracias a esta topología es posible comunicar a un maestro con varios esclavos. Para esto se usa una línea de selección SS (*Slave Select*) a través de la cual se elige el dispositivo con el que se realizará la comunicación.

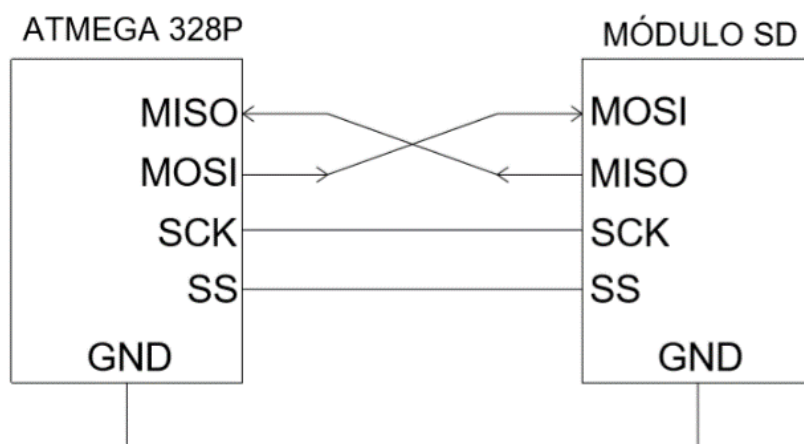


Figura 12: "Conexiones del protocolo SPI".

9.4. Funcionamiento del dispositivo

El funcionamiento del sistema en su conjunto se basará en un esquema cíclico, donde se desarrollará la ejecución del programa de forma estructurada. Este será un esquema como el siguiente:



Figura 13: “Ciclo de funcionamiento del sistema”.

En la primera etapa, el ciclo se corresponde con el calentamiento del sistema. Esto es necesario porque los sensores necesitan un tiempo de calentamiento para que se estabilicen y puedan tomar las medidas correctamente. Por lo tanto, el tiempo de calentamiento del sistema ha sido establecido en función del sensor que más tiempo tarda en estabilizarse.

En la siguiente etapa, se realiza un pequeño chequeo de las conexiones de los puertos seriales, correspondientes a los sensores, y de los puertos SPI, correspondiente al de la memoria flash de la microSD. Si se detectara algún fallo se detiene el sistema para evitar posibles daños a los equipos.

Los puertos seriales utilizados, debido a la falta de estos físicamente en el microcontrolador, se van a crear de forma virtual, es decir, a través de software. Esto es posible gracias a la librería *SoftwareSerial.h*, esta librería está incluida dentro del entorno de programación Arduino IDE, y es capaz de simular un puerto serial Rx-Tx en dos pines digitales cualesquiera del microcontrolador de la siguiente manera:

```
SoftwareSerial CO2Serial(6, 7); // RX , TX CO2
```

Figura 14: “Declaración de UART virtual”.

Una vez el programa realiza el chequeo de los pines, tanto de la UART como del SPI, comienza la ejecución de la comunicación con los sensores a través de los diferentes puertos seriales virtuales configurados. Esta comunicación, al tratarse de una comunicación serial, y gracias a la arquitectura interna de los sensores, de tipo Modbus, basta con enviar el comando deseado especificado en el *datasheet*. Algunos de los comandos utilizados son:

<i>Tabla 4: Comandos utilizados para comunicación con los sensores</i>	
COMANDO	FUNCIÓN
Z<cr><lf>	Lectura y volcado de los datos de medida de concentración del gas del sensor.
H<cr><lf>	Lectura y volcado de los datos de medida de humedad del sensor.
B<cr><lf>	Lectura y volcado de los datos de medida de presión del sensor.
T<cr><lf>	Lectura y volcado de los datos de medida de temperatura del sensor.
c<cr><lf>	Lectura y volcado de los datos de medida de fecha y hora del sensor.

Donde *cr* es el retorno del carro y el *lf* el salto de línea. Por ejemplo, si se desea enviar el comando Z<cr><lf>, basta con enviar al sensor desde el MCU por comunicación UART: “Z 13”, donde “13” es el retorno del carro y el salto de línea ya viene establecido en la función “println”:

```
mySerial.println("Z 13");  
mySerial.listen();
```

Figura 15: “Envío de comando a través del puerto serial”.

De esta manera, el comando deseado es recibido por el sensor y este nos devolverá la información que se corresponda a dicho comando.

Para la lectura de esta información, el microcontrolador escuchará el puerto serial e irá guardando toda su información hasta que el *buffer* quede vacío:

```
while (mySerial.available() > 0)  
{  
    char inByte = mySerial.read();  
    codigo += inByte;  
}
```

Figura 16: “Lectura del puerto serial del sensor”.

Una vez termine el envío por parte del sensor, ya se dispondrá de la información deseada, de esta manera se consigue la comunicación con los sensores y la obtención de toda la información de temperatura, humedad, presión, fecha, hora o concentración del gas del sensor.

Este mismo proceso es análogo para los 3 puertos seriales virtuales para los 3 sensores.

Una vez recogida toda la información, se procesa y se guarda en el formato deseado. A la información se le ha dado formato CSV, este formato es el utilizado en programas de organización como Microsoft Excel. De esta manera, los datos recogidos durante el periodo que esté el sistema midiendo se podrán utilizar y graficar de forma sencilla en este programa o compatibles.

A continuación, toda la información se volcará a la tarjeta microSD. Para esto, se abre un fichero permisos de lectura y escritura para volcar en él todos los datos que se desee:

```
myFile = SD.open(nombreFich, FILE_WRITE);  
if (myFile) {  
    myFile.println(datos);  
    myFile.close();  
}
```

Figura 17: “Volcado de datos a la tarjeta microSD”.

De esta manera, se ha abierto un fichero con nombre “nombreFich”, que ha sido creado a priori con extensión .CSV, en el cual se ha volcado toda la información que se ha recogido y procesado de los sensores.

El nombre del fichero “nombreFich”, será una variable interna del código, que irá variando. De esta manera, se irán creando diferentes ficheros con un número concreto de medidas cada uno. Es decir, si se desea que cada fichero guarde 10 medidas se cambiará el nombre del fichero y se creará uno nuevo cada 10 ciclos. De esta manera, el usuario podrá elegir cómo se organizan los datos. Además, se utiliza como medida de seguridad de cara a la pérdida de información. ya que si algún fichero se daña solo se pierde una parte de la información, y no toda.

Cuando haya finalizado el proceso del volcado de datos, el sistema entra en modo reposo hasta que llegue el momento de volver a tomar medidas. Así se evita que el microcontrolador esté encendido durante periodos innecesarios.

Para ello, el sistema se somete al modo *Power Down*, donde se suspenden todos los procesos y módulos internos de hardware, para reducir el consumo del microcontrolador, a excepción de algunos parámetros para poder despertarlo, como la interrupción del *watchdog*. Para este proyecto es vital poder utilizar la interrupción interna del *watchdog*, pues no hay forma de que una interrupción externa despierte el sistema debido al lugar donde va a estar operando. Para la realización de este estado, se ha utilizado la librería *LowPower.h*, donde permite de una forma sencilla poner el microcontrolador en reposo. Su sintaxis es:

```
LowPower.powerDown(SLEEP_8S, ADC_OFF, BOD_OFF);
```

Figura 18: “Comando para colocar el sistema en modo inactivo”.

A esta función le llega tres parámetros. El primero, es el tiempo del temporizador del perro guardián o *watchdog*. El perro guardián es un temporizador encargado de reiniciar el sistema en caso de bloqueo del código o de despertarlo cuando esté en modo inactivo como interrupción interna. Este temporizador tiene diferentes tiempos de operación, el mayor de ellos es de ocho segundos. Esto es un inconveniente para este proyecto, pues se requiere que el MCU esté inactivo por periodos de tiempo mayores. Por lo tanto, se ha llegado a la coyuntura de dormir y despertar el microcontrolador en bucle, tantas veces sea necesario para llegar al tiempo definido por el usuario. Dado que el sistema solo se despertará durante unos

pocos ciclos de reloj, entre cada iteración del bucle, no se tendrá en cuenta el consumo en esos instantes de tiempo, al tratarse de periodos lo suficientemente pequeños para ser despreciados.

El segundo parámetro de la función permite apagar o dejar encendido los convertidores Analógico/Digitales “ADC_OFF o ADC_ON”, como no se utilizan se apagan. Por último, el tercer parámetro, permite apagar o dejar encendido el circuito *Brown Out Detection*: “BOD_OFF o BOD_ON”, encargado de detectar niveles muy bajos de tensión. Como el sistema está suspendido se ha considerado que no es necesario, por lo que se apaga para reducir el consumo del MCU.

Una vez finalizado las iteraciones de dormir y despertar el MCU, el ciclo vuelve a tomar la ejecución del programa donde se había quedado.

Todo este ciclo de programa se puede consultar en el propio código, en el cual se encuentran más detalladas todas las especificaciones de cada una de las funciones. El código se podrá encontrar en el “ANEXO III: CÓDIGO”

9.5. Validación de datos

Para la validación de los resultados obtenidos a través del dispositivo de medida se ha realizado una serie de pruebas y procedimientos que garantizan la veracidad de los mismos.

- Lectura de los datos de los sensores:

La compañía vendedora de los sensores (CO2Meter) proporciona el software *GasLab* [7] de manera gratuita desde su propia página web. Este programa permite conectar los sensores a través de un USB (Universal Serial Bus). El programa usa una interfaz sencilla, limpia e intuitiva. Una de sus funcionalidades es la recogida de datos leídos por los sensores, volcándolos en un documento *.CSV o incluso graficarlo a tiempo real:



Figura 19: “Mediciones de los sensores en Gaslab”.

En este caso, para garantizar que el microcontrolador estaba recibiendo los valores de las lecturas de los sensores correctamente, se tomaron 30 medidas con cada sensor, dejando 10 segundos entre una lectura y otra. Primero, se hizo con el microcontrolador y después, con el programa proporcionado por la compañía. Por lo tanto, hay que tener en cuenta que las medidas no se realizaron al mismo tiempo, si no con una diferencia de 300 segundos más un pequeño tiempo extra que transcurría entre la desconexión y conexión del sensor. Los datos se graficaron por medio de la aplicación Microsoft Excel.

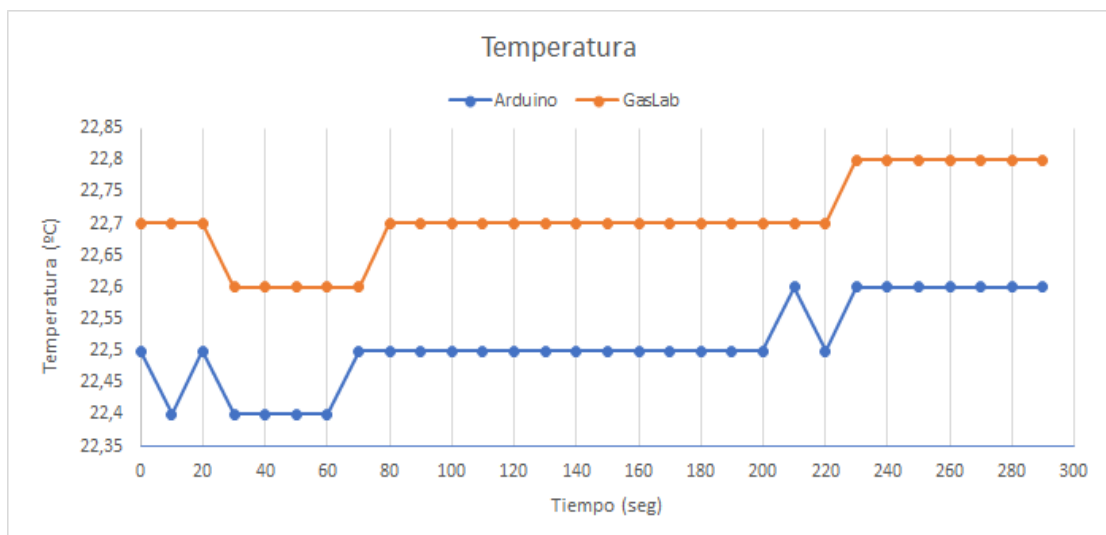


Figura 20: “Comparativa de medición de temperatura entre Arduino y Gaslab”.

En la gráfica de Temperatura, podemos observar cómo las lecturas realizadas por el Arduino y el Gaslab tienen una precisión prácticamente igual. La diferencia entre ellas es de décimas de grado centígrado, por lo que se puede decir que las medidas tomadas tienen una alta fiabilidad. De hecho, si observamos con detenimiento la evolución de la gráfica del Arduino, se puede ver cómo la temperatura aumenta sensiblemente con el tiempo, de la misma manera que pasa después con el programa Gaslab. Luego teniendo en cuenta que las medidas no se realizaron de manera simultánea, se puede llegar a deducir que la precisión entre las temperaturas se puede considerar aún mayor. El incremento de valores es debido a que las lecturas se realizaron al mediodía, cuando la temperatura ambiente estaba aumentando paulatinamente.

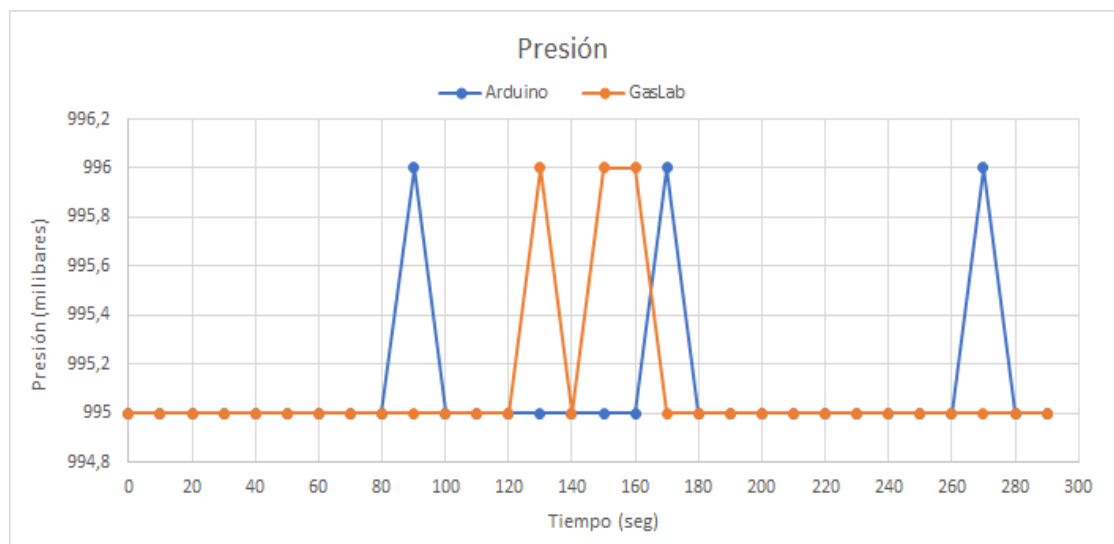


Figura 21: "Comparativa de medición de presión entre Arduino y Gaslab".

En la gráfica de Presión, las lecturas se han mantenido prácticamente constantes, existiendo diferencias despreciables de 1 milibar.

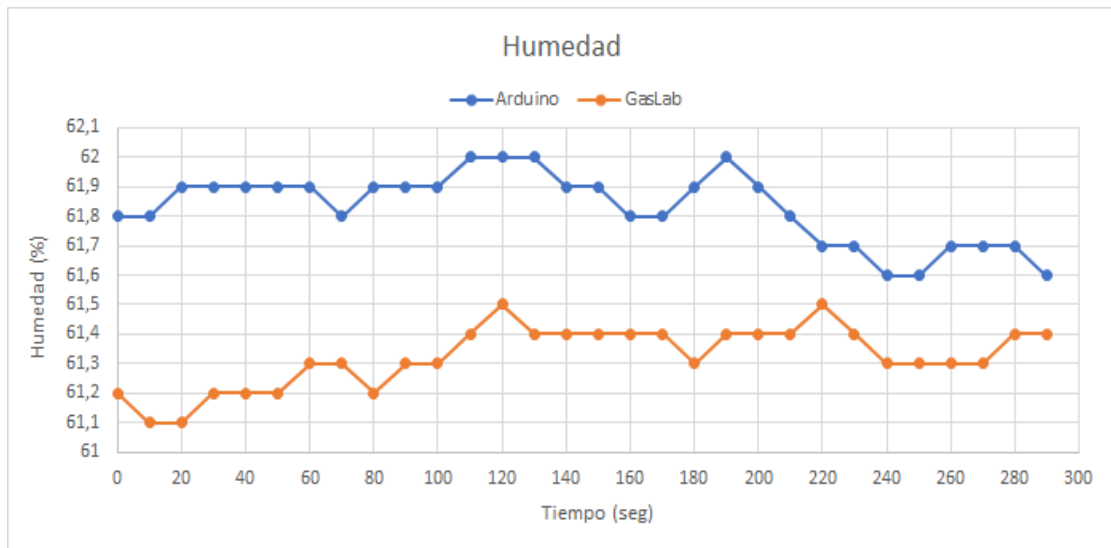


Figura 22: “Comparativa de medición de humedad entre Arduino y Gaslab”.

En la gráfica de la Humedad, las medidas tomadas por el arduino son más elevadas que las del Gaslab, pero siendo siempre inferior a 1%. Teniendo en cuenta que no se realizaron al mismo tiempo y que la diferencia es despreciable, se tomarán las medidas como buenas.

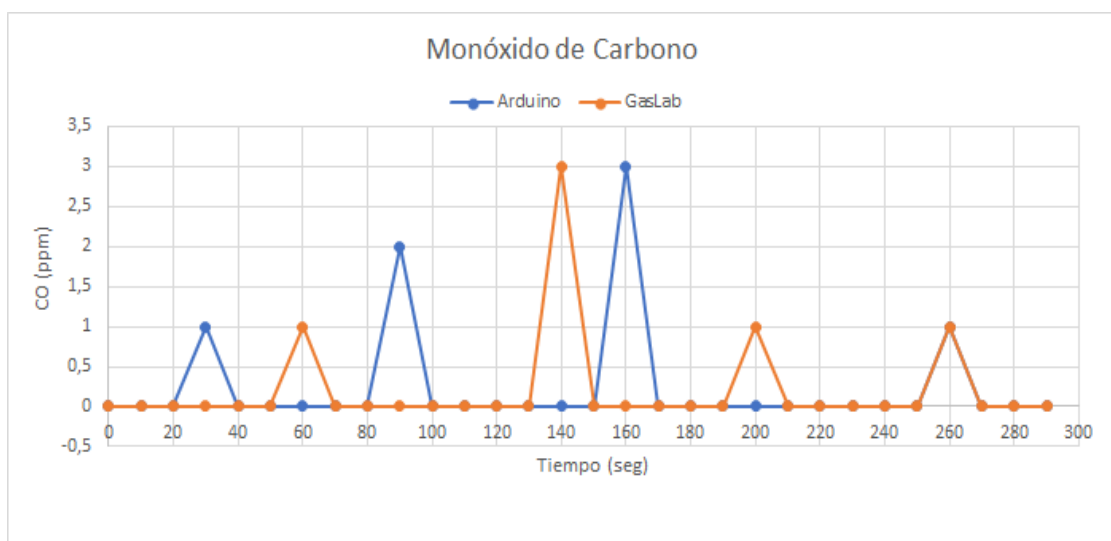


Figura 23: “Comparativa de medición de CO entre Arduino y Gaslab”.

En la gráfica de monóxido de carbono podemos ver que comportamiento del sensor es el mismo, tanto utilizando el Gaslab como el arduino.

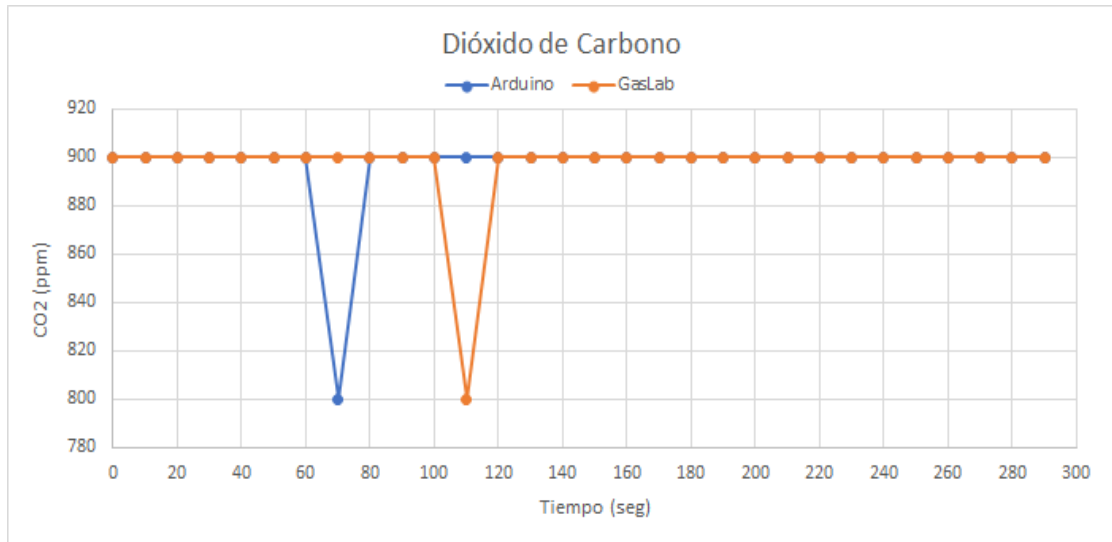


Figura 24: "Comparativa de medición de CO2 entre Arduino y Gaslab".

En la gráfica de dióxido de carbono el funcionamiento del sensor es igual en los dos casos. Las dos simas de la gráfica es debido a la propia resolución del sensor, que toma valores de 100 en 100 ppm.

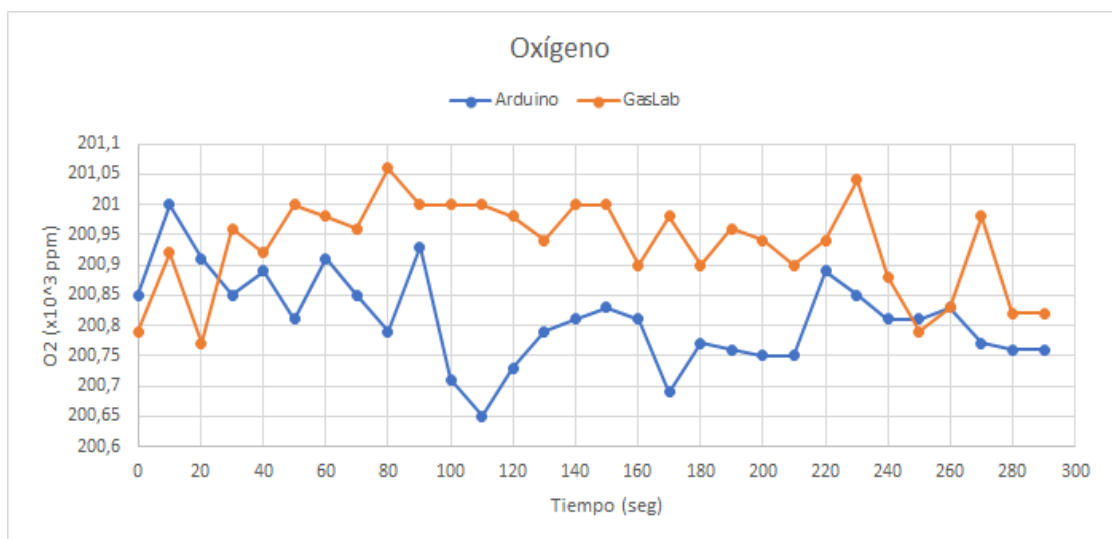


Figura 25: "Comparativa de medición de O2 entre Arduino y Gaslab".

En la gráfica de oxígeno, podemos ver que el comportamiento del sensor es el mismo en los dos casos, habiendo diferencias por debajo de 1000 ppm. Teniendo en cuenta que las medidas tomadas son alrededor de $200 \cdot 10^3$ ppm, podemos despreciar estas disparidades.

Además de estas validaciones, se han hecho pruebas con los sensores durante periodos más largos de tiempo, 6 días en concreto. Tras este periodo se han recogido la información almacenada y se han graficado sus resultados.

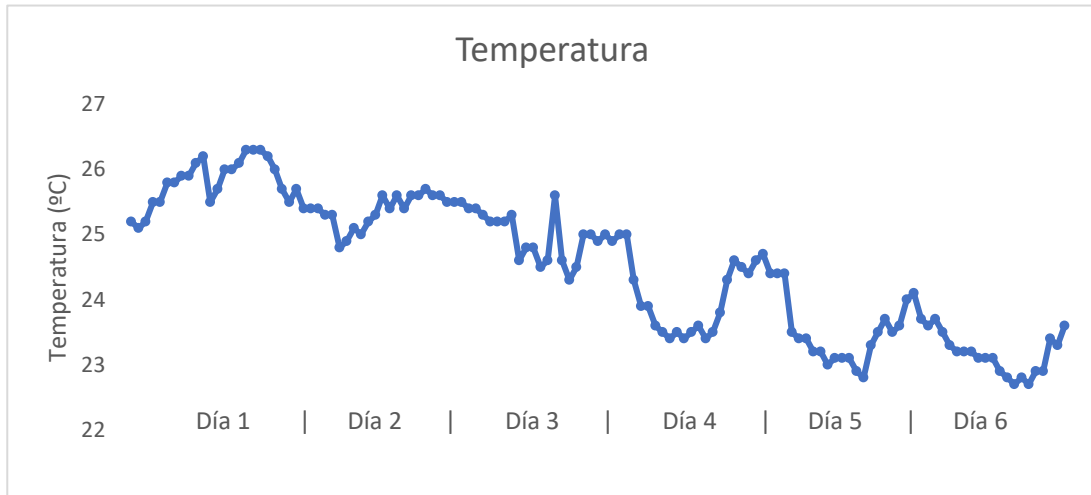


Figura 26: "Medición de temperatura con Arduino durante un periodo de 6 días".

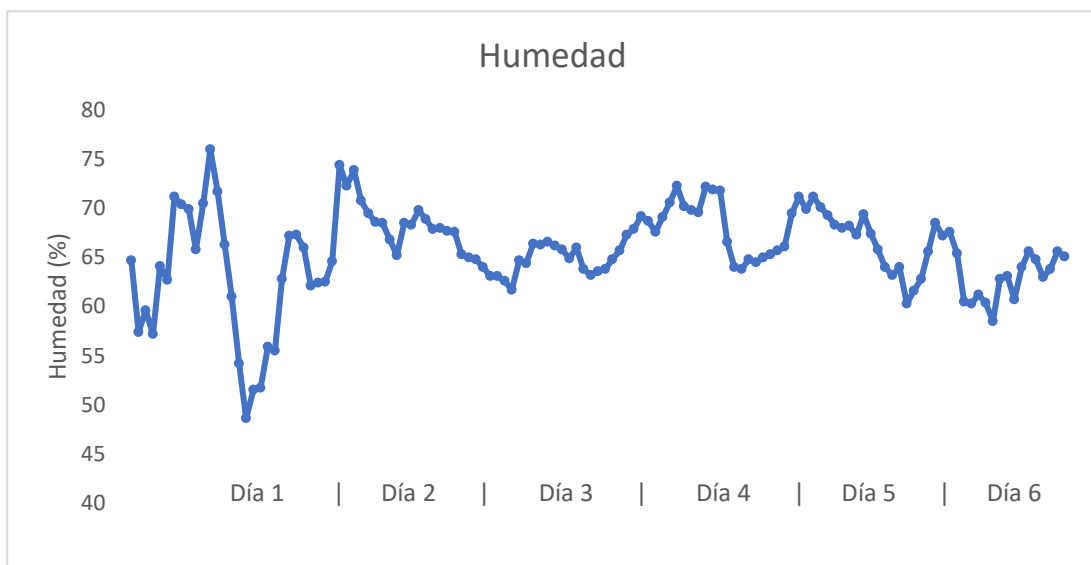


Figura 27: "Medición de humedad con Arduino durante un periodo de 6 días".

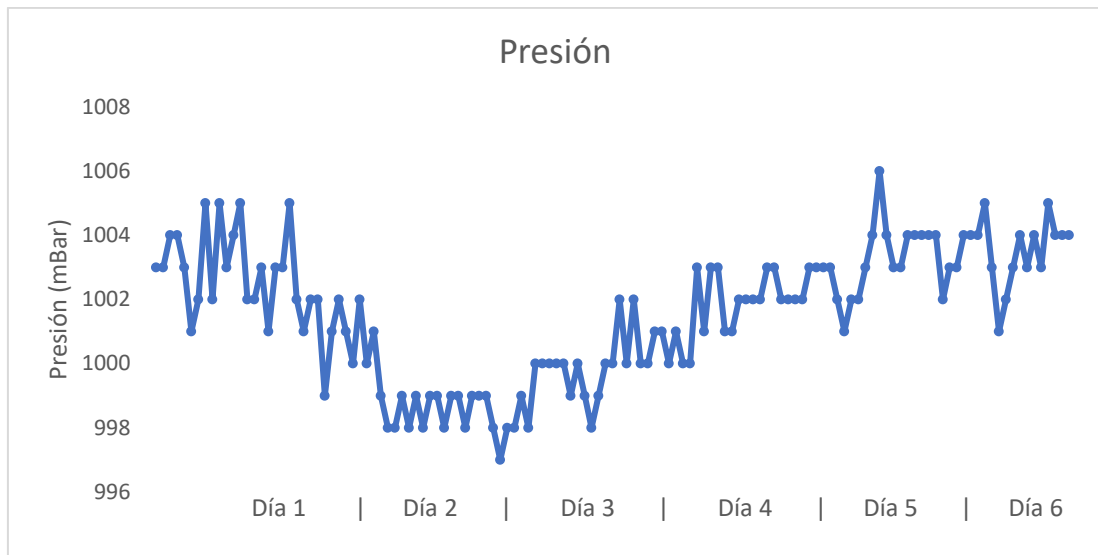


Figura 28: "Medición de presión con Arduino durante un periodo de 6 días".

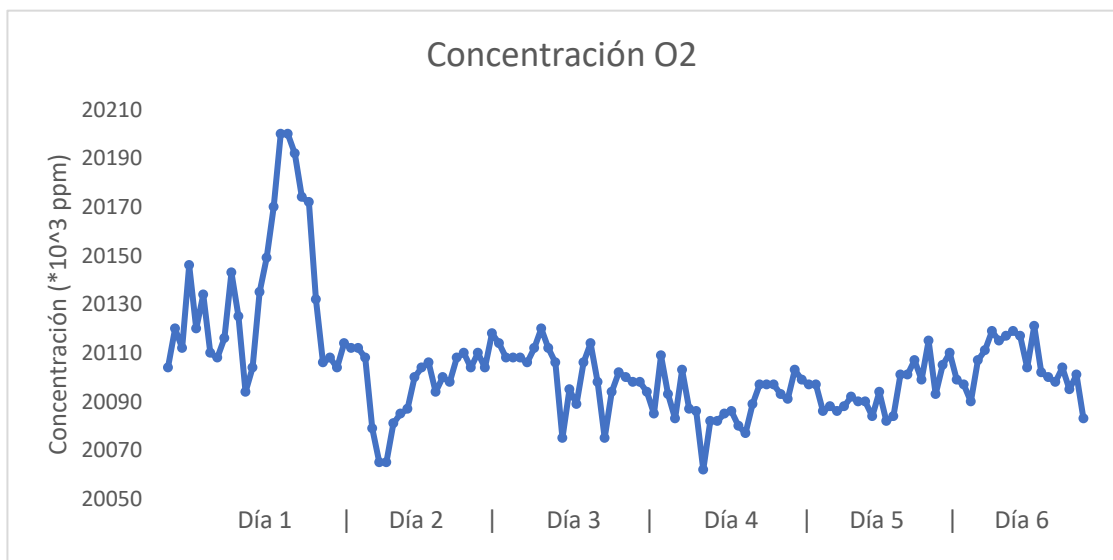


Figura 29: "Medición de concentración de O2 con Arduino durante un periodo de 6 días".

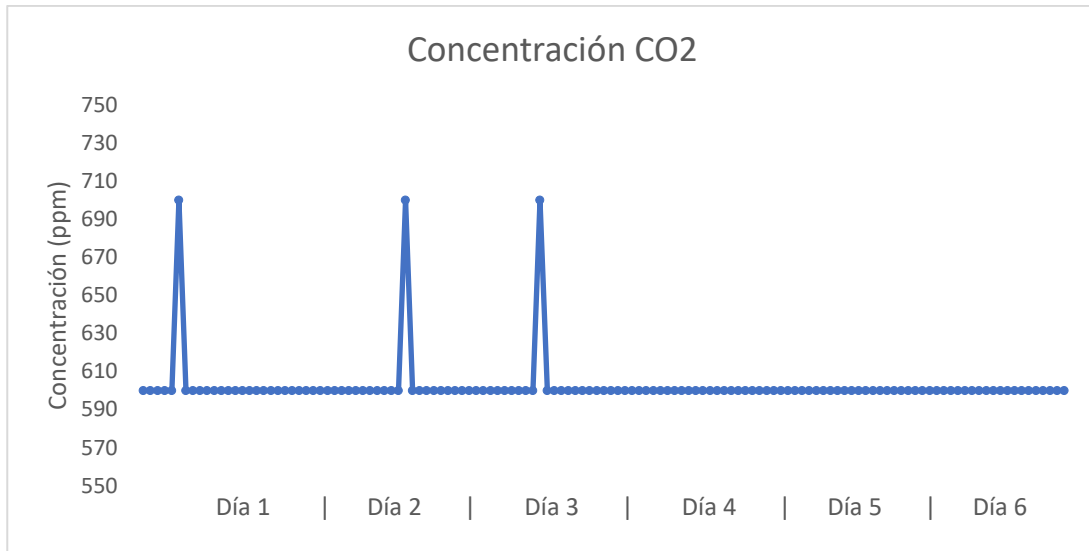


Figura 30: "Medición de concentración de CO2 con Arduino durante un periodo de 6 días".

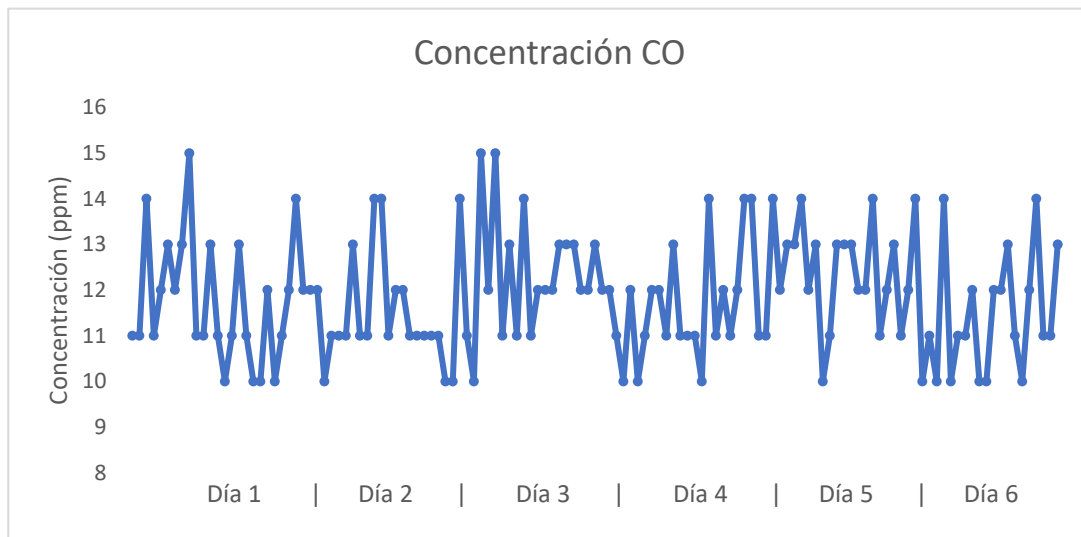


Figura 31: "Medición de concentración de CO con Arduino durante un periodo de 6 días".

9.6. Esquema eléctrico general

Dentro del esquema eléctrico general del sistema final, se pueden diferenciar dos partes. Por un lado, se encontraría la zona de potencia, encargada de gestionar y suministrar la energía de la batería. Esta se compone por las baterías del sistema, encargadas de almacenar toda la energía y los hidrogenadores, encargados de suministrar una pequeña recarga a las baterías, para comprobar cuánta energía son capaces de llegar a ofrecer. Se han realizado pruebas con los hidrogenadores, estos debido a su pequeño tamaño y la poca presión de agua dentro de la galería no son capaces de entregar demasiada potencia. Por esto, para el uso de los

hidrogeneradores como medio de recarga de baterías, es necesaria la implementación de un circuito BMS (*Battery Master Switch*). Su implementación es necesaria por dos motivos, primero, este circuito consta de una impedancia de entrada muy alta, esto es importante, ya que al no disponer de un gran caudal de agua la potencia entregada por los hidrogeneradores será pequeña. De esta manera con una impedancia de entrada alta se evitará que se desplome la tensión cuando se solicite mucha corriente. De esta forma, se consigue una tensión estable a la salida de los hidrogeneradores, con una baja corriente. También, es necesario el BMS para evitar que se produzca un cambio en el sentido de la corriente. En el caso de los hidrogeneradores, esto podría producirse si bajara mucho el caudal de agua y, por consiguiente, su potencia suministrada. Al pasar esto, los hidrogeneradores no solo dejarían de aportar energía al sistema, sino que se comportaría como motores consumiéndola.

De esta manera, con este componente se consigue gestionar, tanto la recarga de las baterías, como el suministro de energía al sistema

Por el otro lado, el esquema eléctrico recuadrado a la izquierda del plano compone la parte integrada dentro del circuito impreso. En esta parte del esquema, se ven reflejadas todas las conexiones eléctricas necesarias para su funcionamiento. En primer lugar, se pasa la alimentación de 6V proveniente del circuito de potencia, por un regulador de tensión lineal, para ajustarla a la tensión requerida tanto por el microcontrolador como por los sensores. Para las conexiones eléctricas del microcontrolador, basta con la conexión directa de los 5V regulados a sus pines VCC, AVCC Y AREF, además se ha de colocar esta misma tensión en el pin RESET por medio de una resistencia. Ya que el reinicio del microcontrolador es a través de lógica negada, será necesario colocar en paralelo en el pin RESET, un pulsador conectado directamente a tierra. Gracias a esto, se reiniciará el sistema cada vez que se pulse dicho botón. La conexión eléctrica de los sensores es simple, todos los sensores están conectados a la alimentación estable de 5V. Pero sus caminos de retorno a tierra están condicionados por el estado de un transistor de efecto campo, controlado por una salida digital del microcontrolador. De esta forma, se consigue con una simple señal digital, cambiar el estado de los transistores de corte a saturación, cerrando el circuito y encendiendo los sensores. Por último, el conexionado de la tarjeta microSD, es análogo al descrito anteriormente para los sensores, con la única diferencia de que será necesaria una nueva regulación del voltaje, para bajarlo hasta los 3.3V de tensión de trabajo de la tarjeta. El esquema completo, con todas las conexiones eléctricas del sistema se puede encontrar en el “ANEXO I: PLANOS” N°P.1.

9.7. Consumo del dispositivo

Tras el montaje final del circuito, utilizando como cerebro el ATMEGA-328P, se ha comprobado que el consumo global del sistema es bastante eficiente, entorno a 1Ah para el periodo de 6 meses operando sin interrupciones, estableciendo un periodo de muestro de una hora. Si se hubiera realizado el montaje directamente con la placa de desarrollo Arduino UNO, el consumo durante el mismo periodo de tiempo hubiera sido de 91.6Ah. Esto habría conllevado un sobrecoste importante a la hora de la compra de baterías y, además, dificultaría mucho su introducción en la galería debido al tamaño de estas.

Todos los cálculos referentes a la potencia consumida, tanto los cálculos del prototipo con Arduino UNO, como los del diseño final con el ATMEGA328P. Se pueden encontrar en el “ANEXO II: CÁLCULOS DE POTENCIA”.

9.8. Diseño del circuito impreso

9.8.1. Programa de diseño

El software utilizado para diseñar el circuito impreso ha sido KiCad [8]. KiCad es un software de código abierto para la automatización de diseño electrónico (EDA, Electronic Design Automation). El software soporta programas para el diseño de esquemáticos (Schematic capture) y el diseño de PCB (PCB layout). Es un programa muy versátil y utilizado por muchos usuarios, por lo que cuenta con una gran cantidad de librerías que puedes descargar desde su página web. Además, al ser un software de código abierto muchos usuarios tienen librerías propias que son accesibles para todos.



Figura 32: “KiCad”.

9.8.2. Esquemático y diseño de la PCB

El esquemático del diseño del dispositivo de medida se encuentra adjuntado en el *ANEXO I: PLANOS*, N°. Plano: 2. Se han utilizado etiquetas para procurar que el esquema sea lo más organizado posible. Posteriormente, al pasar el netlist al diseño de PCB, interpreta dichas etiquetas como conexiones entre ellas.

La placa de circuito impreso tiene unas dimensiones bastante reducidas, de 78x63 mm. La PCB es bicapa o DSB (Double-Sided Printed Circuit Boards), es decir, la interconexión de los componentes se ha hecho entre las dos caras del sustrato.

Respecto a las reglas de diseño:

- Se incorpora agujero de sujeción en las cuatro esquinas con diámetro de 3 milímetro de taladrado.
- El ancho de pista ha sido como máximo 1 milímetro y como mínimo hasta 0.5 milímetros.
- El trazado de pistas se ha realizado de manera manual, de forma que se ha evitado las bifurcaciones en la medida de lo posible y no se han producido ángulos de 90°. Además, se ha estrechado o ensanchado el tamaño de la pista cuando ha sido necesario de manera sutil, para evitar cualquier tipo de corte de cobre cuando se vaya a realizar la fabricación de la placa. A su vez, se ha procurado ensanchar lo máximo posible la pista de VCC.
- Separación entre pistas global track-track de 0.5 mm.
- Separación entre pista y pad al menos de 0.254 mm.
- Se ha incorporado un condensador electrolítico a la salida del conector para evitar el posible rizado de la fuente de alimentación, y condensadores cerámicos en los reguladores de tensión para evitar que entren en auto-oscilación, tal y como recomienda el fabricante en el datasheet.
- Se ha procurado que la placa sea lo más compacta posible, para no desperdiciar material y para que sea más sencillo trabajar con ella.
- Se ha creado un área de cobre que funciona como plano de tierra en cada capa, de esta manera favorecemos la señal de las pistas, evitando el acoplamiento entre ellas. Además, este plano de cobre contribuye a un tiempo de fabricación menor, pues no se deberá de esperar a la eliminación de la superficie de cobre. Asimismo, la placa estará menos tiempo expuesta al ácido, por lo que la calidad de las pistas aumentará.

Por otro lado, en la siguiente imagen se muestra la cara superior del diseño de la PCB en 3D:

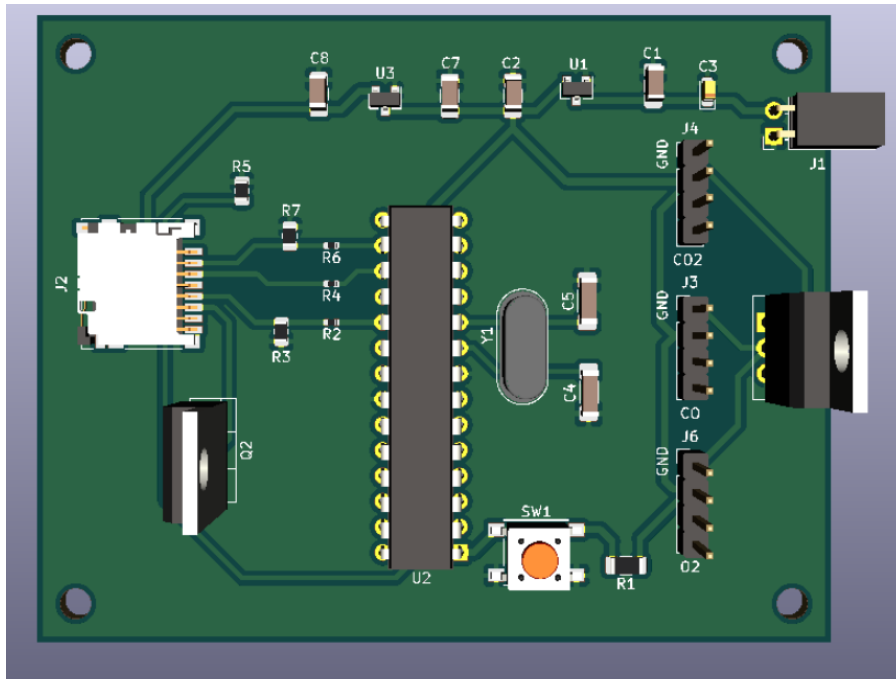


Figura 33: "Frontal de la PCB en 3D".

En el diseño físico de la PCB, el MCU no irá soldado a la placa, a diferencia de cómo se ve en el prototipado 3D. En realidad, se va a poner un zócalo mismas dimensiones del ATMEGA 328 - P, para poder introducir o extraer el MCU tantas veces el usuario desee, así como poder cambiarlo por otro. Así se facilita la programación del chip, que se basa únicamente en extraerlo para después introducirlo en una placa de desarrollo Arduino UNO y programarlo en el entorno de programación Arduino IDE.

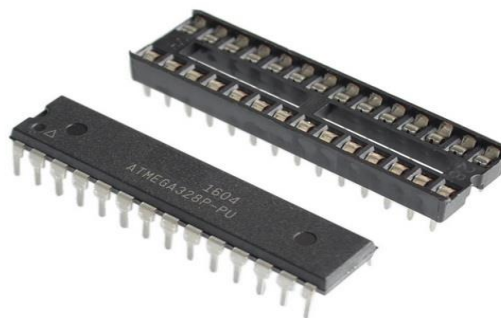


Figura 34: "Zócalo para el ATMEGA328P en la PCB".

Por otro lado, no se desea que los sensores vayan soldados a la placa, pues se deberán de situar en zonas estratégicas para que puedan tomar las medidas pertinentes, esto está explicado en el apartado 10. *Encapsulado*. En contraposición, se han incorporado tres conectores machos verticales 1x4 de tipo *pinHeader*, con 2.54 mm de separación entre pines, que es la misma configuración de pines físicos que tienen los sensores. De esta manera, resulta sencillo y práctico conectar los sensores con conectores hembras, pudiendo así cambiarlos una vez se acabe su ciclo de vida por otros, sin necesidad de estar desoldando la PCB.

Conector que conecta el sensor y la placa:



Figura 35: “Conectores para los sensores y la PCB”.

Por otro lado, los fotolitos generados para la fabricación de la PCB, y el prototipado 3D desde la cara superior, cara inferior y de lado, se encuentran en los planos los que sea en el “ANEXO I: PLANOS” N°P. 3, 4 y 5.

10. Encapsulado

Para proteger el dispositivo de medida y la fuente de alimentación, se van a encapsular en una caja. Se ha seleccionado una caja de *RS Pro*, desde *RS Components*, de la marca comercial de *Electrocomponents plc*. Tiene una estructura de acero templado y con recubrimiento electrostático de epoxi y poliéster, con nivel de protección IP 66 y dimensiones 300x200x150 milímetros, suficientes para introducir el dispositivo de medida y al menos dos baterías. Con este nivel de protección garantizamos la no entrada de partículas pequeñas ni

suciedad y ofrece una gran resistencia al agua y un nivel de protección de chorros potentes de agua. Además, cuenta con IK 10 contra impactos mecánicos.

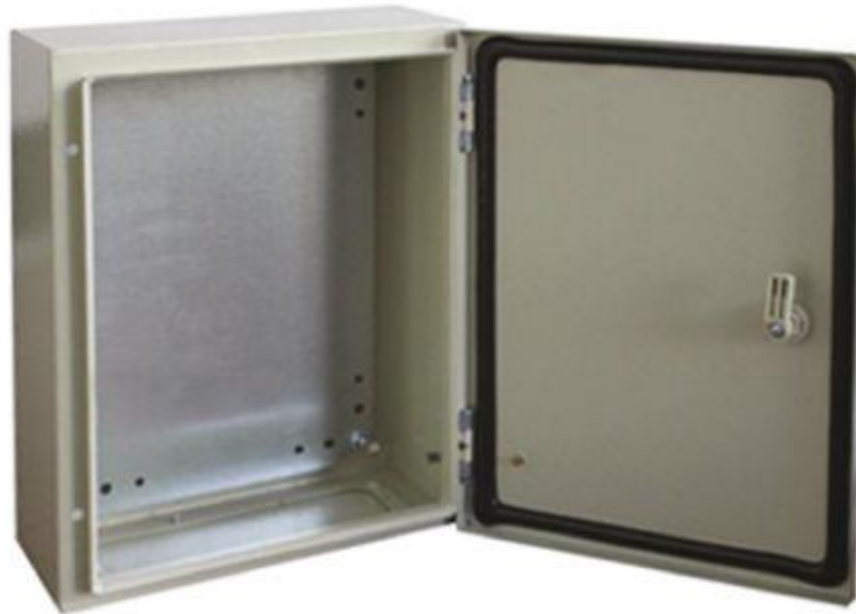


Figura 36: "Caja para encapsulado del sistema".

La caja se puede utilizar tanto en el suelo como empotrada en la pared, donde incluye una placa posterior de acero zincado, una placa de montaje para la puerta, prensaestopas y una junta de estanquidad. Cuenta con el certificado de conformidad RoHS (*Restriction of Hazardous Substances*).

Por lo tanto, una vez introducido la/s batería/s dentro de la caja, se introduce la PCB, utilizando los cuatro agujeros de sujeción de las esquinas para empotrarlo en la superficie interior e interior de la caja. Respecto a los sensores, se va a realizar tres bifurcaciones en la superficie exterior de la tapa de la caja, de diámetro igual al diámetro del sensor (sin la electrónica, únicamente la zona necesaria para que el sensor pueda leer los gases del exterior de la caja) más dos milímetros para incorporar una junta tórica, para asegurar la estanquidad de fluidos al interior de la caja. Es decir, un diámetro de 20.1 más 2 milímetros.

De la misma manera que con la PCB, los sensores se pueden colocar a la altura suficiente en el interior de la caja para realizar las medidas utilizando los cuatro agujeros de sujeción que tiene en sus esquinas, aplicando posteriormente silicona o sellante estanquidad en las zonas que se realicen las bifurcaciones, o bien utilizando un soporte desde el interior de la caja.

11. Presupuesto

11.1. Introducción

En este apartado se explicará de una forma más detallada el presupuesto de este prototipo. Incluyendo en este el coste de todos los materiales necesarios para su fabricación y los costes de toda la mano de obra invertida en ella.

11.2. Costes materiales

El coste material del proyecto estará formado por los costes unitarios de cada uno de los componentes utilizados y su cantidad. Los costes materiales se van a dividir en costes materiales de fabricación de la PCB, “Tabla 5”, y costes del sistema de baterías y su encapsulado “Tabla 6”. Se ha hecho de esta manera por si fuera necesario reemplazar alguna parte del sistema en un futuro.

DESCRIPCIÓN	CANTIDAD	PRECIO UNITARIO	PRECIO TOTAL
Sensor O2	1	289,90 €	289,90 €
Sensor CO	1	289,90 €	289,90 €
Sensor CO2	1	289,90 €	289,90 €
ATMEGA328P	1	2,26 €	2,26 €
MicroSD	1	1,22 €	1,22 €
REGULADOR MCP1703A-3302	1	0,50 €	0,50 €
REGULADOR MCP1703A-5002	1	0,50 €	0,50 €
Transistor MOSFET IRL540N	2	0,50 €	1,00 €
Condensadores	8	0,15 €	1,20 €
Resistencias	7	0,15 €	1,05 €
Cristal de cuarzo 16 MHz	1	1,10 €	1,10 €
Pulsador	1	0,25 €	0,25 €
Conector	1	1,05 €	1,05 €
Fabricación de PCB	1	33,53 €	33,53 €
TOTAL	28		913,36 €

DESCRIPCIÓN	CANTIDAD	PRECIO UNITARIO	PRECIO TOTAL
Caja de encapsulado IP67	1	33,53 €	33,53 €

Baterías	2	31,06 €	62,12 €
BMS	1	6,50 €	6,50 €
Cables	4	0,31 €	1,24 €
TOTAL	8		103,39 €

11.3. Costes de mano de obra

Todos los procesos necesarios para la realización del prototipo se pueden dividir en cinco bloques: programación, optimización del sistema, fabricación de la PCB, encapsulado del sistema y documentación. Conociendo el número de horas invertidas en cada una de estas tareas, se puede calcular el coste total de la mano de obra “Tabla 7”.

<i>Tabla 7: Coste de las baterías y encapsulado</i>			
<i>DESCRIPCIÓN</i>	<i>CANTIDAD</i> <i>(horas)</i>	<i>PRECIO UNITARIO</i>	<i>PRECIO TOTAL</i>
Programación	130	25 €	3250 €
Optimización del sistema	70	30 €	2100 €
Fabricación de PCB	25	20 €	500 €
Encapsulado	25	20 €	500 €
Documentación	50	25 €	1250 €
TOTAL	300		7600 €

11.4. Coste de ejecución material

Para concluir este apartado, se dará el coste total de la realización del proyecto sumando los costes materiales con los de mano de obra. Ver en “Tabla 8”.

<i>Tabla 8: Coste de ejecución material</i>	
<i>DESCRIPCIÓN</i>	<i>PRECIO TOTAL</i>
Costes de la PCB	913,36 €
Costes de baterías y encapsulado	103,39 €
Costes mano de obra	7600 €
<i>TOTAL</i>	8616,75 €

11.5. Gastos generales

Además de los costes de ejecución material, hay que añadir al presupuesto total del proyecto los costes generales, provenientes de gastos en papeleo y secretaría, estos se suelen calcular a través del coste de ejecución material, suponiendo aproximadamente un 16% de dicho coste.

<i>Tabla 9: Gastos generales</i>	
<i>DESCRIPCIÓN</i>	<i>PRECIO TOTAL</i>
Gastos generales	1378,68 €

11.6. Beneficios industriales

Al igual que en todos los proyectos, el objetivo por parte de los desarrolladores, a parte de subsanar una necesidad, es la recompensación económica por su trabajo. En este se va a aplicar como beneficio un 6% del coste de ejecución material.

<i>Tabla 10: Beneficios industriales</i>	
<i>DESCRIPCIÓN</i>	<i>PRECIO TOTAL</i>
Beneficios industriales	517 €

11.7. Impuestos

Una vez realizados todos los cálculos de gastos y beneficios del proyecto, hay que añadir a esta cifra sus correspondientes impuestos a pagar. En este caso, se han de aplicar los impuestos pertenecientes a la Comunidad de Canarias, en la que se aplica el IGIC (Impuesto General Indirecto Canario), este se corresponde con un 7%.

<i>Tabla 11: Impuestos (IGIC)</i>	
<i>DESCRIPCIÓN</i>	<i>PRECIO TOTAL</i>
Impuestos	730,87 €

11.8. Presupuesto de contrata

A continuación, se pasa a dar el presupuesto total de contrata.

<i>Tabla 12: Presupuesto de contrata</i>	
<i>DESCRIPCIÓN</i>	<i>PRECIO TOTAL</i>
Coste de ejecución material	8616,75 €
Gastos generales	1378,68 €
Beneficios industriales	517 €
Impuestos	730,87 €

TOTAL	11243,30 €
-------	------------

12. Conclusions

A system has been designed to collect data on temperature, humidity, pressure and gas concentrations of O₂, CO and CO₂. The achievements made are:

- Use a microcontroller with less physical features than necessary, making modifications through software to create them virtually.
- To have a very efficient consumption, getting to be below the milliamperes in the periods of inactivity. Due to this consumption, it has been possible to obtain long periods of work, overcome the year of autonomy with a sampling of measurements every hour with only one battery. So, requirements of the client have been surpassed, who asks for at least the autonomy of six months with the same time of sampling using up to two batteries.
- Simple system to manipulate by any user, being able to have all the information stored in a memory card. In addition, the user can change the sampling period by modifying a single variable in the code.
- The information is stored and structured so that it can be read directly in data management programs, such as Microsoft Excel.
- The device is able to work at high temperatures and humidity, splashes, water jets and mechanical impacts until the established protection levels.

On the other hand, the incorporation of renewable energies is proposed as an improvement, taking advantage of the perpetuity of running water in the gallery.

13. Bibliografía

- [1] Universidad de Burgos. (2015). Normativa electrónica. José M. Cámara.
<https://riubu.ubu.es/bitstream/handle/10259/3589/Normativa.pdf?sequence=1&isAllowed=y>

- [2] Dogan, I., (2007). Programación de microcontroladores PIC, Barcelona, España: Editorial
- [3] Hambley Allan R., (2001). Electrónica, Barcelona, España: Editorial
- [4] ¿Que es Arduino? (2019, 3 diciembre). Arduino.cl - Compra tu Arduino en Línea. <https://arduino.cl/que-es-arduino/>
- [5] Arduino - Software. (2015, 14 abril). Arduino. <https://www.arduino.cc/en/main/software>
- [6] PROTOCOLOS DE COMUNICACIÓN SPI/I2C/UART/UART. (2018, 6 abril). COMUNICACIÓN SERIAL. <https://udaytdf.wixsite.com/udaytdf/post/protocolos-de-comunicaci%C3%B3n-spi-i2c-uart-uart>
- [7] Software & Documentation Downloads. (2019). CO2 Meter. <https://www.co2meter.com/pages/downloads>
- [8] *KiCad EDA. (2020). KiCad EDA. KiCad Website. <https://kicad-pcb.org/>*
- [9] Blog Artelijon. (2018). El Proyecto Electrónico. <http://blog.artelijon.com/ortal/files/2007/09/ut1-el-proyecto-electronico.pdf>



**Universidad
de La Laguna**

ESCUELA SUPERIOR DE INGENIERÍA Y TECNOLOGÍA

Trabajo de Fin de Grado

DISEÑO DE DISPOSITIVO DE BAJO CONSUMO PARA LA
LECTURA DE GASES DENTRO DE UNA GALERÍA

Pliego de Condiciones

Titulación: Grado en Ingeniería Electrónica Industrial y Automática

Autores: Ezequiel Cortés Martín

Alberto Doro García

Tutores: Jose Francisco Gómez González

Santiago Torres Álvarez

Contenido

1. Campo de aplicación.....	52
2. Condiciones generales.....	52
2.1. Marco jurídico	52
2.2. Régimen de intervención.....	52
2.3. Accidentes de trabajo y daños a terceros	52
2.4. Régimen y certificados	53
2.5. Rescisión del contrato	53
3. Condiciones de materiales y equipos	53
4. Condiciones de ejecución.....	54
5. Condiciones económicas	55
5.1. Garantías.....	55
5.2. Precios contradictorios.....	55
5.3. Pagos.....	55
5.4. Indemnización por retrasos o defectos	56

1. Campo de aplicación

Este sistema está diseñado para la toma de medidas de gases dentro de espacios cerrados de difícil acceso. Dadas las condiciones extremas de estos lugares, el sistema cuenta con protecciones contra salpicaduras de agua y resistencia a altas temperaturas.

Está diseñado para disponer de una autonomía mínima de 1 año, utilizando la batería recomendada por el fabricante. Por lo que no requerirá de ningún mantenimiento durante este periodo. Tras este periodo, toda la información se encontrará almacenada en una memoria microSD.

2. Condiciones generales

2.1 Marco jurídico

La ejecución de este proyecto se llevará a cabo teniendo en cuenta las leyes vigentes del estado, siendo esta responsabilidad de la empresa, y siendo la responsable del cumplimiento de estas por parte del contratista.

2.2 Régimen de intervención

Si el contratista no diera cumplimiento a sus obligaciones legales estipuladas en el contrato, la empresa será la responsable de dar órdenes a este para su cumplimiento. El contratista dispondrá de un plazo estipulado por la empresa para la subsanación de todas estas obligaciones legales.

Si estos incumplimientos fueran recurrentes o por el contrario se tratara de incumplimientos graves, la empresa tendrá el derecho a la rescisión del contrato de forma unilateral, sin necesidad de indemnización al contratista.

2.3 Accidentes de trabajo y daños a terceros.

El contratista será el único responsable de la seguridad de todos sus trabajadores durante la ejecución del proyecto. Tendrá la obligación de comprobar el cumplimiento de las leyes de seguridad y salud de sus trabajadores, y será el responsable de cualquier accidente del cual queda exonerada la empresa.

Además, si tras la venta de cualquier producto por parte de la empresa, algún usuario sufriera algún daño por él, debido a negligencias a la hora de su fabricación por parte del contratista, este será el único responsable.

2.4 Permisos y certificados

El contratista será el encargado de la solicitud de todos los permisos y certificados necesario para la ejecución del proyecto. Este deberá contar con toda esta documentación y deberá aportársela a la empresa si esta lo requiriera en cualquier momento.

2.5 Rescisión del contrato

Las siguientes acciones podrán ser causantes de la rescisión del contrato por parte de la empresa:

- Demora de los plazos de entrega.
- Incumplimientos del marco normativo.
- Incumplimiento de responsabilidades.
- Fallos en los productos por negligencias del contratista.
- Modificaciones del presupuesto con incrementos superiores al 15%.

Por consiguiente, las acciones por parte de la empresa que puedan causar que el contratista pueda rescindir el contrato:

- Incumplimiento en los pagos.
- Plazos de entrega irrealizables.

Todas estas rescisiones de contrato conllevarán indemnizaciones dependiendo del tipo de incumplimiento, y será responsabilidad de ambas partes llegar a un consenso de esta.

3. Condiciones de materiales y equipos

Para la elaboración de este proyecto, el contratista dispondrá de todos los datos necesarios de los materiales que componen el diseño, además de todas las especificaciones estructurales del sistema a construir. El contratista deberá ceñirse a los materiales especificados. De no ser así, estaría incumpliendo el contrato y este podría ser rescindido. Para la realización

de cualquier cambio de los materiales especificados en el diseño, se deberá realizar una propuesta y esta ser ratificada por la empresa.

Por último, si el contratista realizara una subcontratación, deberá facilitar toda la información de los materiales y sus especificaciones a esta. Además, dicha subcontratación deberá demostrar su cualificación ante la empresa y ser aprobada por ella.

4. Condiciones de ejecución

El usuario, antes de colocar el sistema en su ubicación para la toma de medidas, deberá asegurarse de la ausencia de agua corriente en esta. Además, deberá asegurar que sea una ubicación estable y no haya riesgo de desprendimiento.

Los orificios situados en la parte superior del sistema han de estar en todo momento libres y sin ningún obstáculo encima de ellos. Estos orificios son los encargados de la entrada de aire a los sensores para la medida de gases, al ser obstaculizados, el sistema no podrá recoger de forma correcta las medidas.

Cuando se desee abrir el encapsulado del sistema, para el cambio de baterías o la recogida de la información. El usuario deberá estar atento en todo momento de que no entre agua en su interior bajo ningún concepto. Además, tras realizar las labores que desee, deberá volver a colocar la tapa del sistema y cerrarlo debidamente, asegurándose de que esté correctamente sellado y no haya posibilidad de la filtración de agua a su interior.

A la hora de realizar un mantenimiento, solo será necesaria la apertura del encapsulado, la retirada de la memoria microSD y su sustitución por una nueva y el cambio de baterías. De esta manera el sistema ya estará listo para volver a operar.

Cada año, se debe realizar un mantenimiento preventivo. Se debe revisar que el encapsulado no tenga signos de corrosión o desgaste. Si el encapsulado está dañado, se deberá de cambiar inmediatamente. Asimismo, se debe de revisar el estado de los conectores, tanto de los sensores como de las baterías.

5. Condiciones económicas

5.1 Garantías

El sistema de medida tendrá una garantía de dos años desde su primer encendido por el cliente. El tiempo que el sistema esté en garantía, todos los desperfectos en sus componentes serán cubiertos por la empresa, sin ningún coste para el cliente.

Además, si el sistema fallara durante su periodo de aislamiento en la galería, el cliente sería indemnizado a razón de la gravedad del fallo.

La garantía podrá quedar anulada si el dispositivo ha sido dañado o tiene muestras evidentes de una manipulación incorrecta. Además, también se podrá anular la garantía si no se presenta prueba de compra o si el número de serie se ha modificado.

5.2 Precios contradictorios

Si en algún caso, durante la ejecución del proyecto por parte del contratista, fuera necesario el cambio del precio de algún componente. Se procederá a su estudio y ratificación por parte de la empresa. La empresa estará obligada a conciliar el nuevo precio con el contratista.

La revisión de precios solo se podrá hacer efectiva si, tras la firma del contrato, el presupuesto total se incrementara en, al menos, un 10%.

5.3 Pagos

El pago del presupuesto, por parte de la empresa hacia el contratista, se realizará tras la firma del contrato y antes del comienzo de la ejecución del proyecto.

Si en algún caso, la empresa no deseara seguir con el proyecto, por circunstancias ajenas al contratista, esta perderá el presupuesto invertido.

De la misma manera, si el contratista, ya comenzado el proyecto, deseara rescindir el contrato, por circunstancias ajenas a la empresa, este deberá devolver el presupuesto integro, además de una indemnización proporcionada a los inconvenientes generados a la empresa.

5.4 Indemnización por retrasos o defectos.

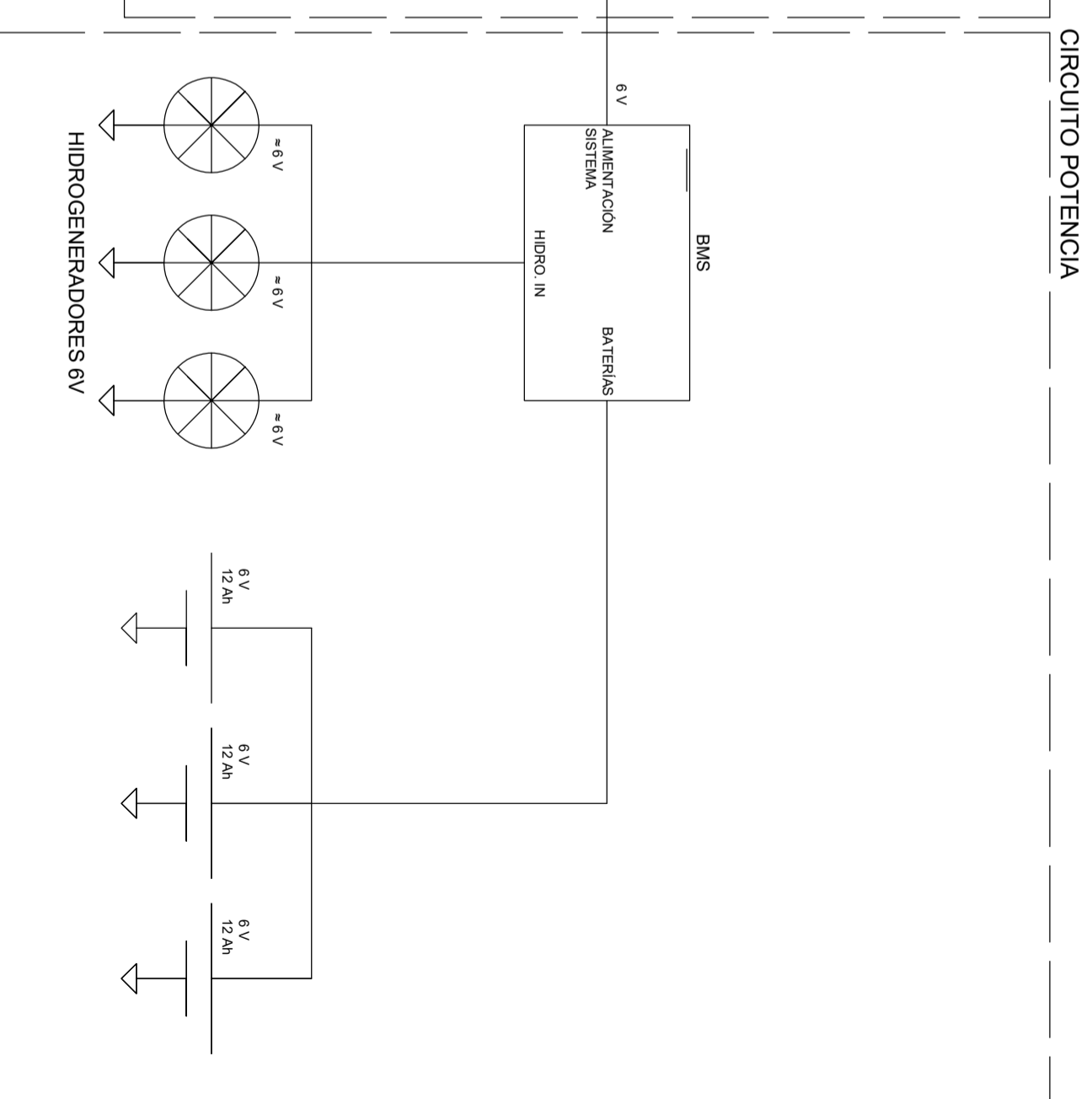
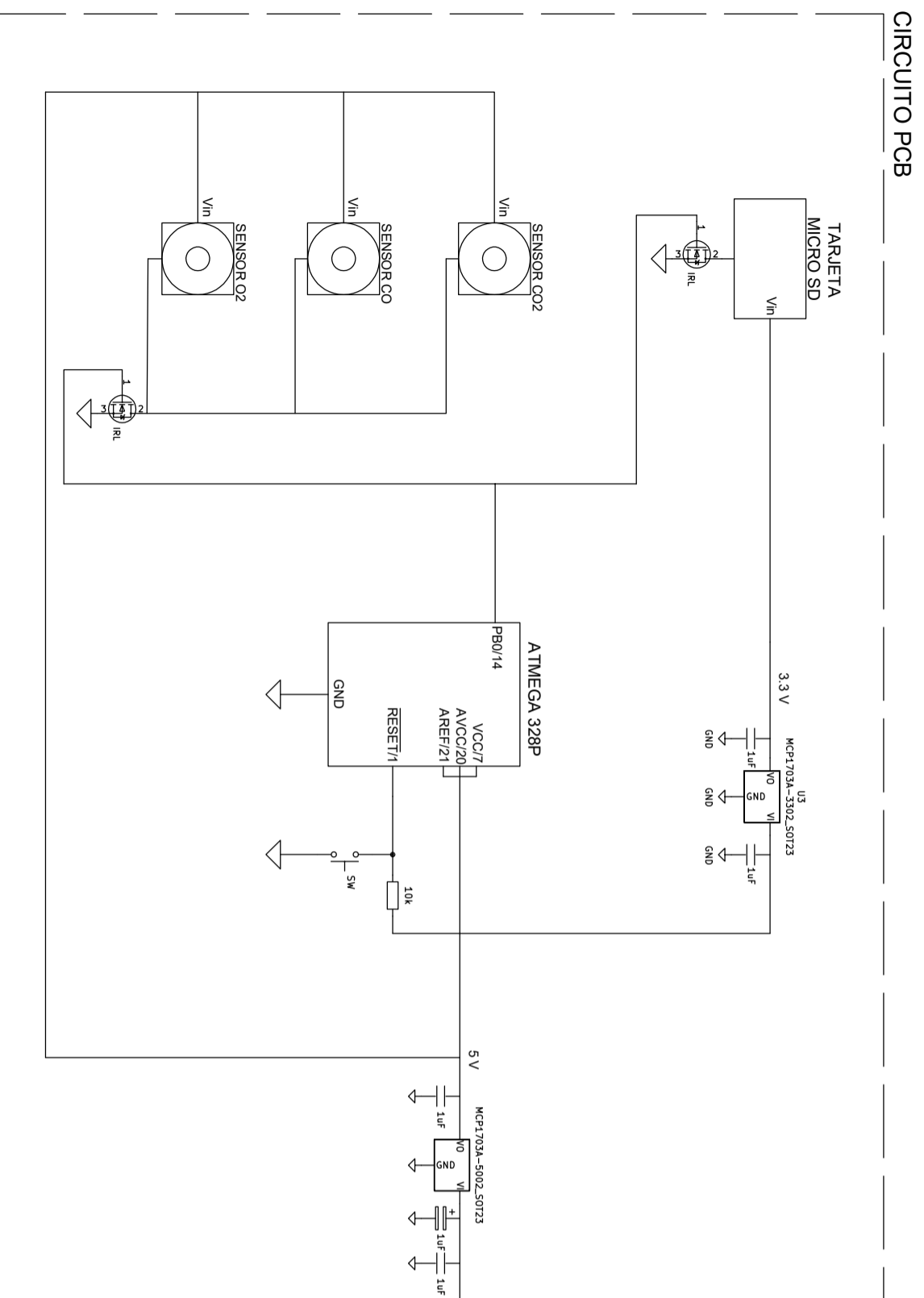
Si en algún caso, el contratista no entregara el proyecto en las fechas estipuladas, este deberá indemnizar a la empresa con una cantidad proporcionada a la duración del retraso. Esta indemnización deberá ser abonada a la hora de la entrega del proyecto terminado.


Además, si tras la entrega del proyecto, este no cubriera las necesidades estipuladas por la empresa, el contratista deberá indemnizar a la empresa con una cantidad proporcional a la gravedad del defecto del proyecto entregado [9].

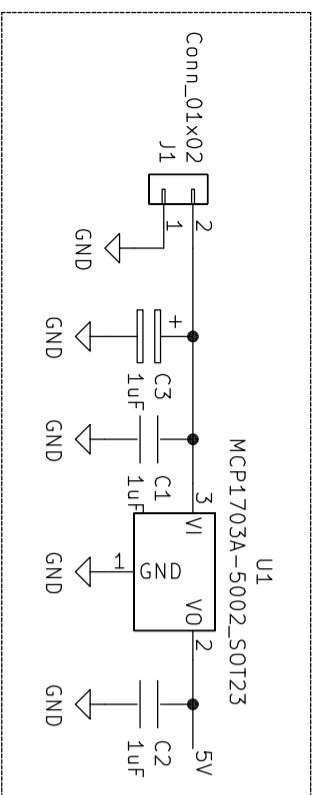
ANEXOS

ANEXO I

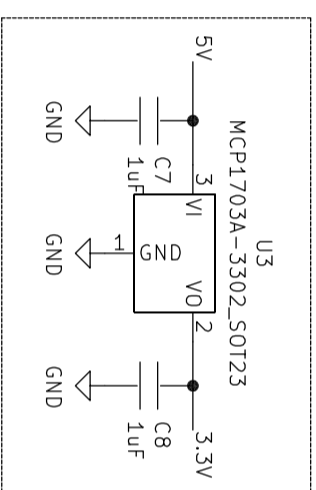
PLANOS



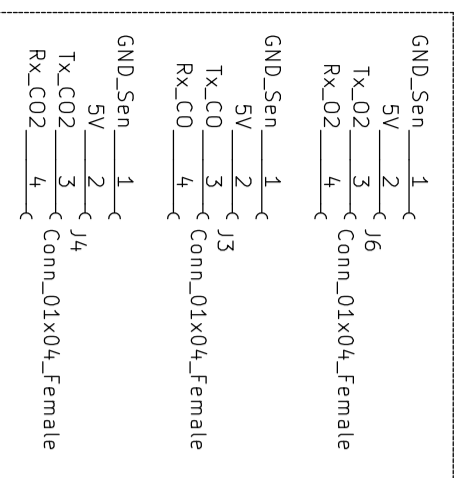
TRABAJO FIN DE GRADO			
Fecha	27 de Junio de 2020	 Universidad de La Laguna	Grado Ingeniería Electrónica Industrial Universidad de La Laguna
Autor	EZEQUIEL CORTÉS MARTÍN		
Autor	ALBERTO DORO GARCÍA		
Id. s. normas	UNE-EN-DIN	ESQUEMA ELÉCTRICO GENERAL	
S/E		Nº P. : 1	Nom. Arch. : TRABAJO FIN DE GRADO



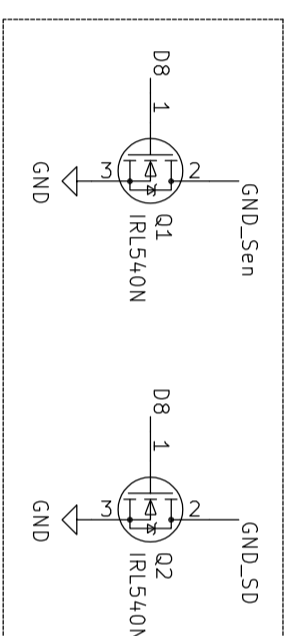
Regulador de 5V



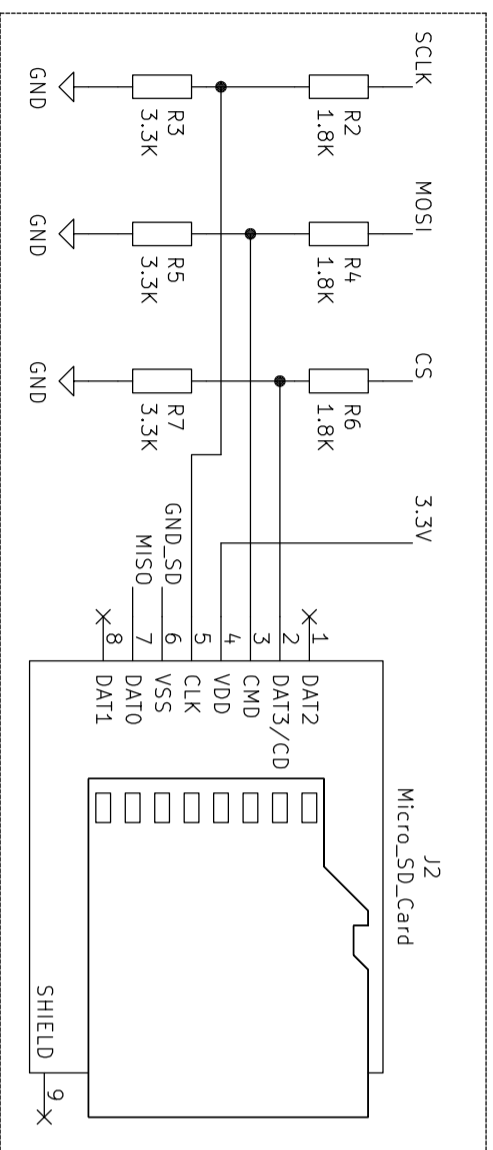
Regulador de 3.3V



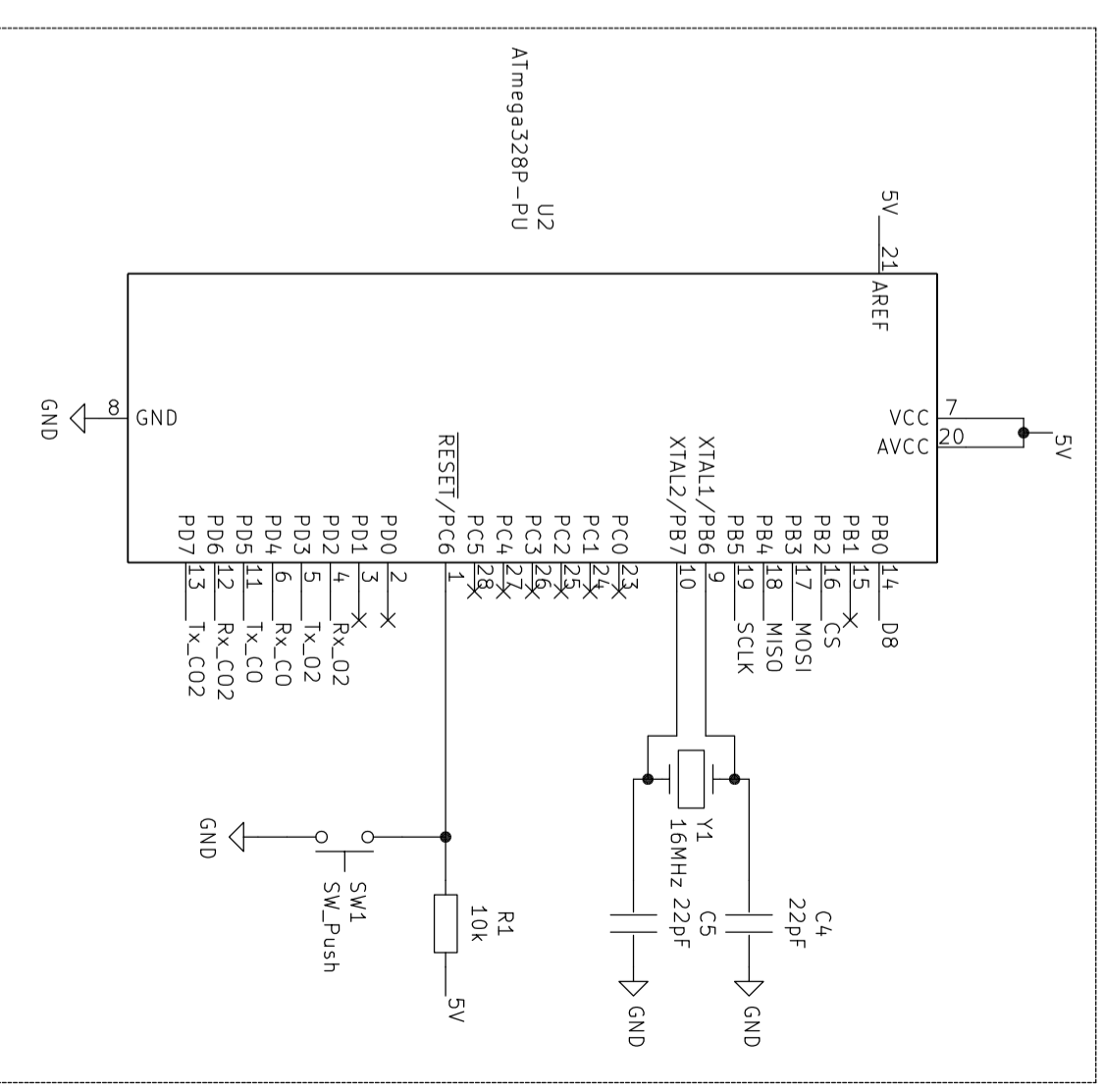
Conectores de los sensores



Mosfet IRL540N



Socket tarjeta micro SD

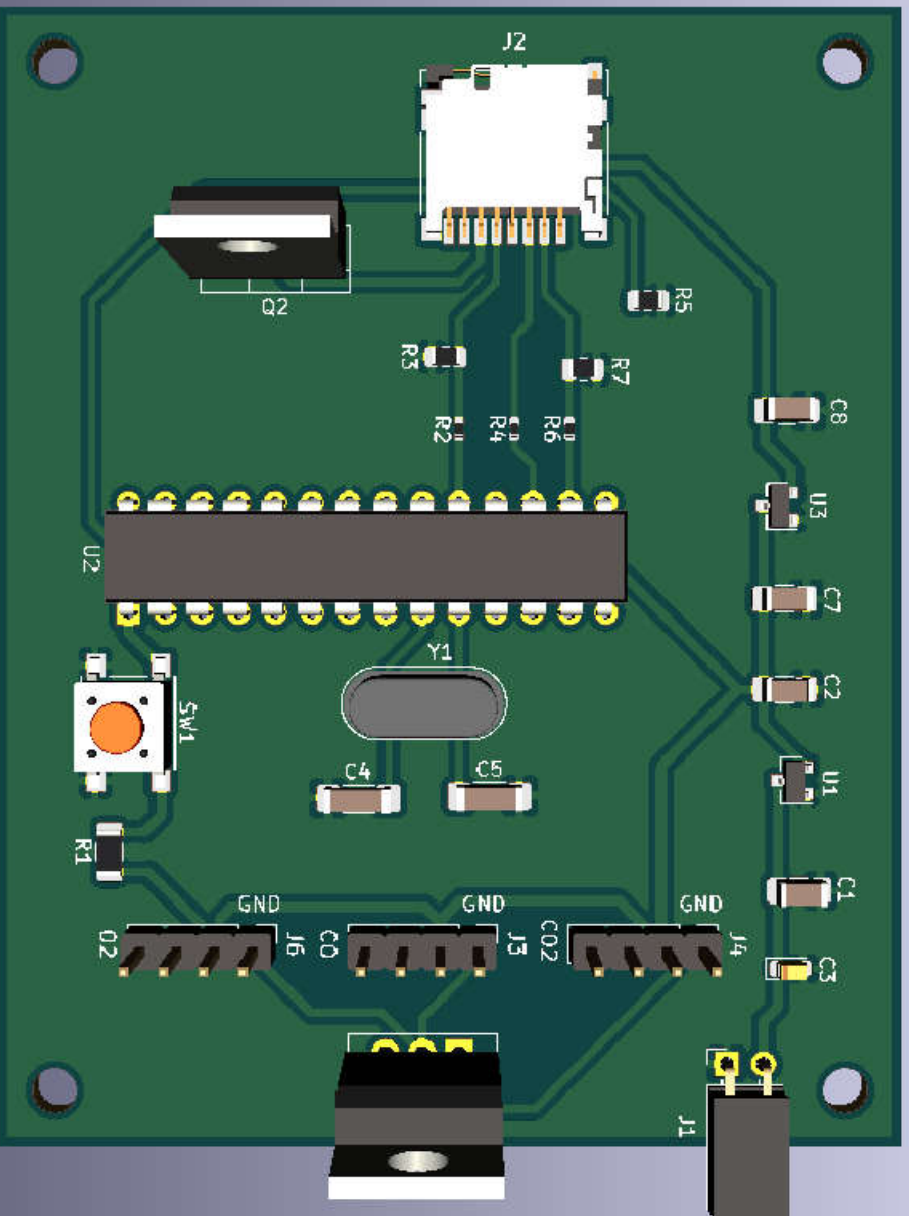


ATMEGA 328P

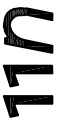
TRABAJO FIN DE GRADO			
Fecha	27 de Junio de 2020		
Autor	EZEQUIEL CORTÉS MARTIN		
Autor	ALBERTO DORO GARCIA		
Id. s. normas	UNE-EN-DIN		
ESCALA: S/E		ESQUEMÁTICO DE PCB	Nº P.: 2 Nom.Arch: TRABAJO FIN DE GRADO

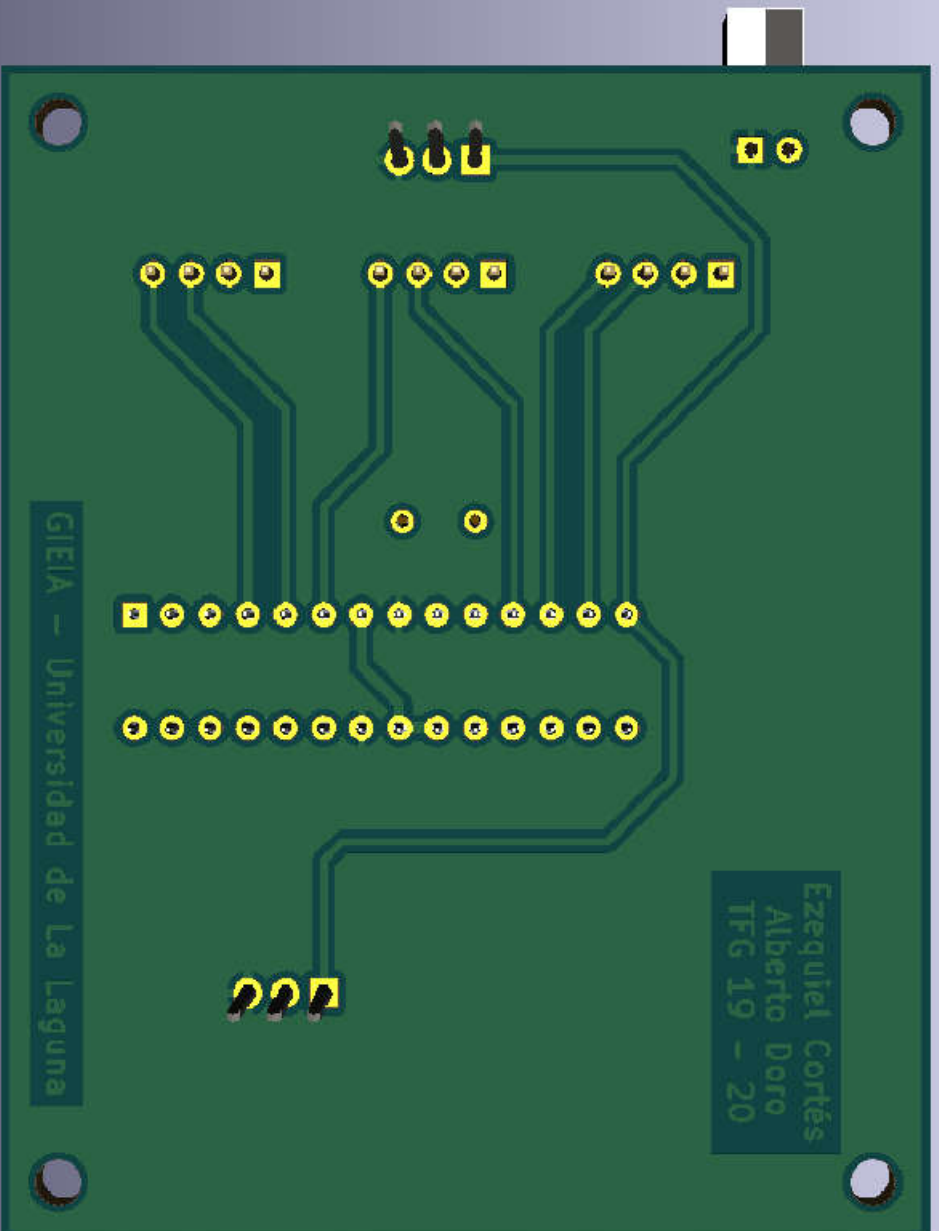
U11
Universidad de La Laguna

ES.I.C.I.
Grado Ingeniería Electrónica Industrial
Universidad de La Laguna




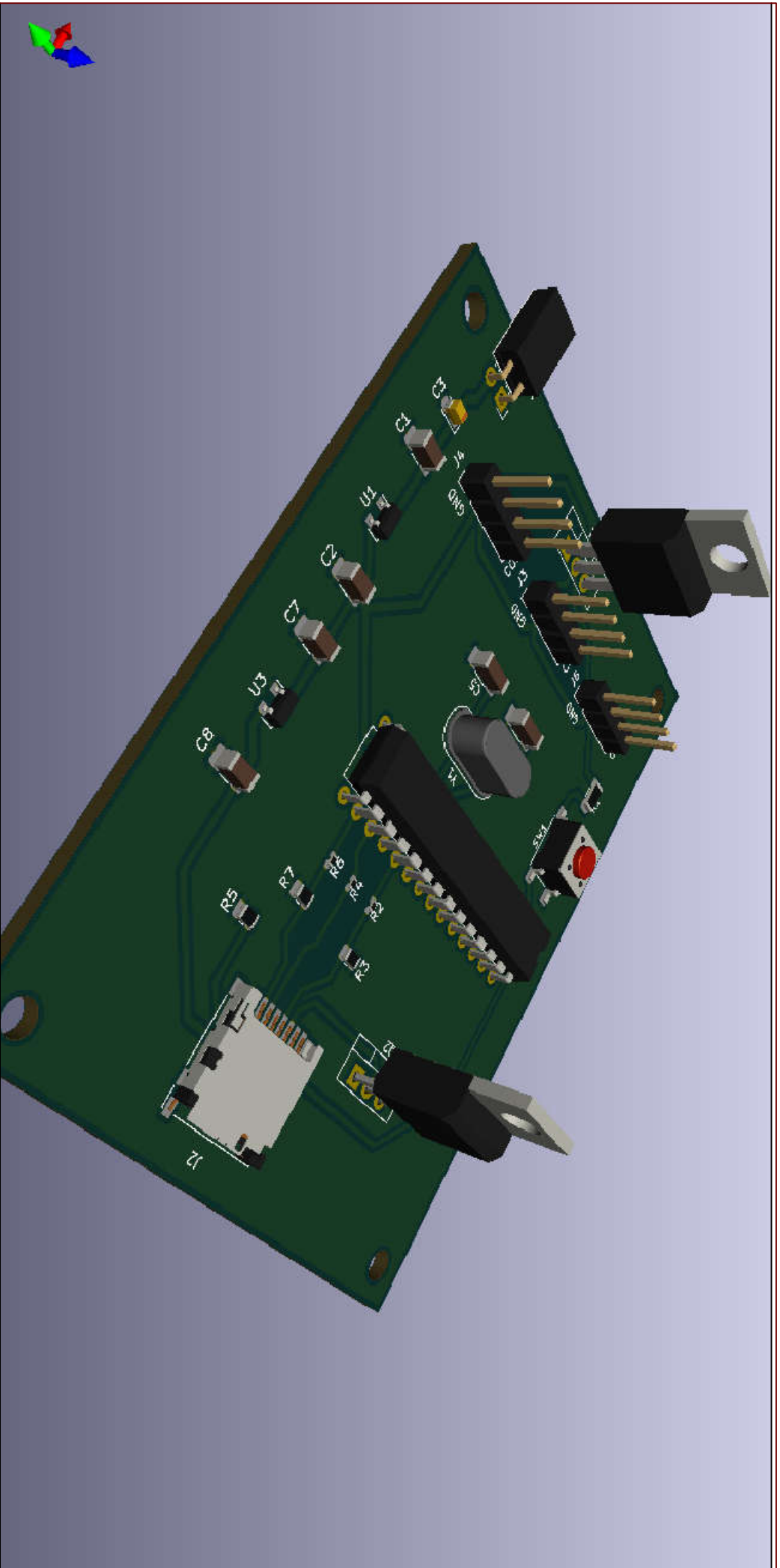
TRABAJO FIN DE GRADO

TRABAJO FIN DE GRADO		ES.I.C.I.	
Fecha	27 de junio de 2020	 Grado Ingeniería Electrónica Industrial y A. Universidad de La Laguna	
Autor:	EZEQUIEL CORTÉS MARTÍN		
Autor:	ALBERTO DORO GARCÍA		
Id. s. normas	UNE-EN-DIN	Nº P.: 3 Nom. Arch.: TRABAJO FIN DE GRADO	
ESCALA:	S/E		
PROTOTIPO 3D: CARA SUPERIOR			



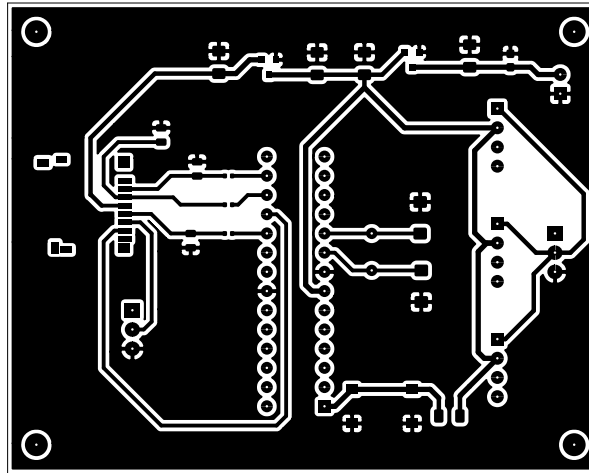
TRABAJO FIN DE GRADO

Fecha		27 de junio de 2020	
Autor:		EZEQUIEL CORTÉS MARTÍN	
Autor:		ALBERTO DORO GARCÍA	
Id. s. normas		UNE-EN-DIN	
ESCALA:		S/E	
PROTOTIPO 3D: CARA INFERIOR		 Universidad de La Laguna	
ES.I.C.I.		Grado Ingeniería Electrónica Industrial y A.	
Universidad de La Laguna			
Nº P. : 4			
Nom. Arch: Protipo_3D_PCB_TFG			



TRABAJO FIN DE GRADO

Fecha		27 de junio de 2020	
Autor:		EZEQUIEL CORTES MARTIN	
Autor:		ALBERTO DORO GARCIA	
Id. s. normas		UNE-EN-DIN	
ESCALA: S/E		PROTOTIPO 3D: VISTA DE LADO	
		<p>ULL Universidad de La Laguna</p> <p>Grado Ingeniería Electrónica Industrial y A. Universidad de La Laguna</p>	
Nº P.: 5		Nom.Arch: Protipo_3D_PCB_TFG	



Alberto Doro García
Ezequiel Cortés Martín
Departamento de Ingeniería Industrial de la Universidad de La Laguna

Sheet:
File: TFG_PLACA_19-20.kicad_pcb

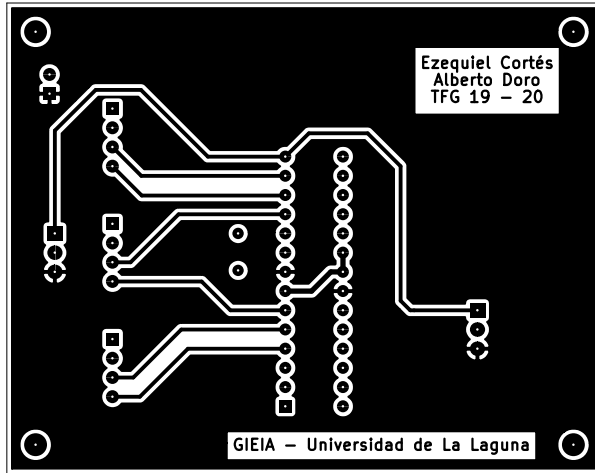
Title: Fotolito de la capa de cobre superior

Size: A4 Date: 2020-06-26

KiCad E.D.A. kicad (5.1.6)-1

Rev: N°. Plano: 6

Id: 1/1

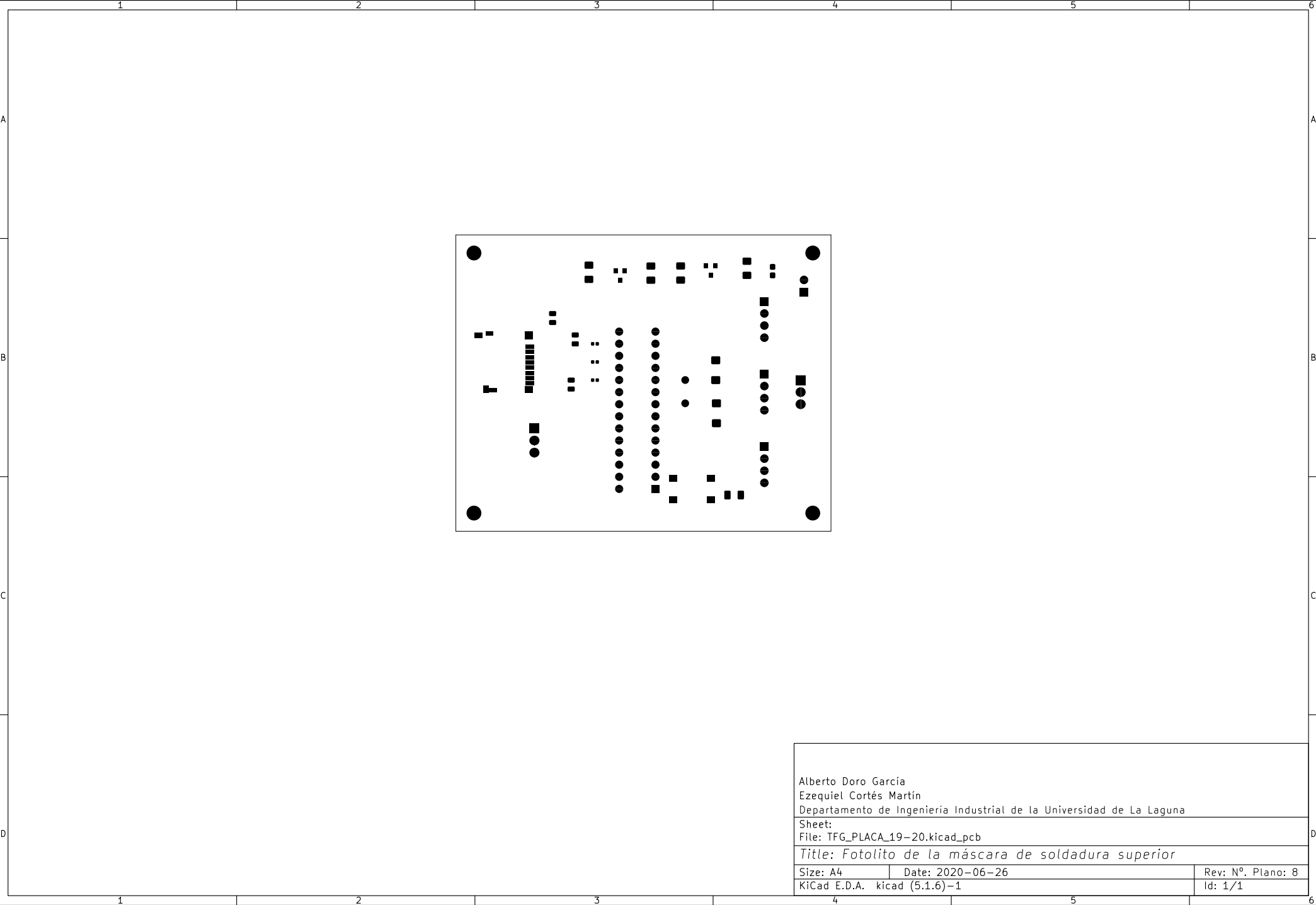


Alberto Doro García
 Ezequiel Cortés Martín
 Departamento de Ingeniería Industrial de la Universidad de La Laguna

Sheet:
 File: TFG_PLACA_19-20.kicad_pcb

Title: Fotolito de la capa de cobre inferior

Size: A4	Date: 2020-06-26	Rev: N°. Plano: 7
KiCad E.D.A. kicad (5.1.6)-1		Id: 1/1

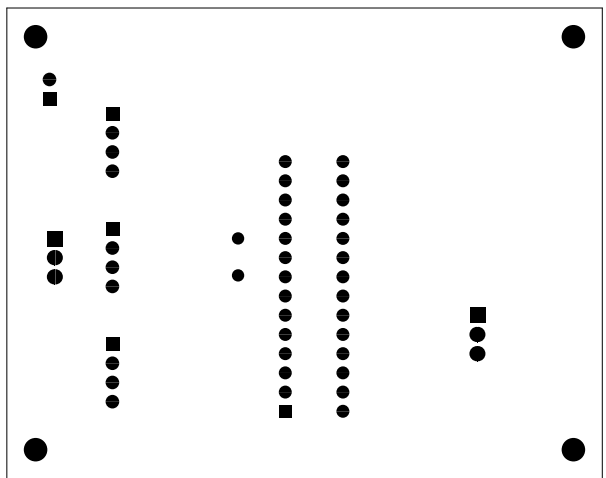


Alberto Doro García
Ezequiel Cortés Martín
Departamento de Ingeniería Industrial de la Universidad de La Laguna

Sheet:
File: TFG_PLACA_19-20.kicad_pcb

Title: Fitolito de la máscara de soldadura superior

Size: A4	Date: 2020-06-26	Rev: N°. Plano: 8
KiCad E.D.A. kicad (5.1.6)-1		Id: 1/1

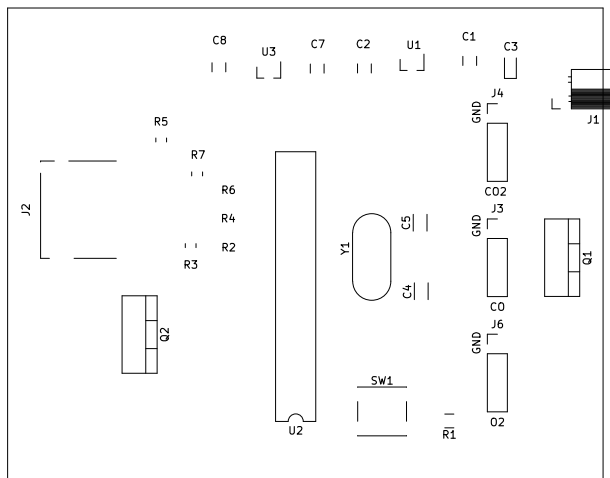


Alberto Doro García
Ezequiel Cortés Martín
Departamento de Ingeniería Industrial de la Universidad de La Laguna

Sheet:
File: TFG_PLACA_19-20.kicad_pcb

Title: Fotolito de la máscara de soldadura inferior

Size: A4	Date: 2020-06-26	Rev: N°. Plano: 9
KiCad E.D.A. kicad (5.1.6)-1		Id: 1/1



Alberto Doro García
 Ezequiel Cortés Martín
 Departamento de Ingeniería Industrial de la Universidad de La Laguna

Sheet:	
File: TFG_PLACA_19-20.kicad_pcb	
Title: <i>Fotolito de la serigrafía</i>	
Size: A4	Date: 2020-06-26
KiCad E.D.A. kicad (5.1.6)-1	Rev: N°. Plano: 10
	Id: 1/1

ANEXO II

CÁLCULOS ELÉCTRICOS

ANEXO II: CÁLCULOS ELÉCTRICOS

A continuación, se van a realizar los cálculos de las potencias consumidas por el sistema y la energía necesaria para que el sistema sea capaz de ser autónomo durante un tiempo especificado. Estos cálculos se van a realizar desde varios puntos de vista. En primer lugar, se van a realizar las estimaciones teóricas del consumo del sistema con los componentes seleccionados anteriormente, pero utilizando el entrenador Arduino UNO.

Una vez realizado los cálculos teóricos, se van a calcular las medidas experimentales del sistema, para ver si realmente los cálculos realizados se ajustan a la realidad del sistema. Por último, ya con todos los cálculos teóricos verificados de forma experimental, se va a pasar a realizar una estimación del número de baterías necesarias para que el sistema sea capaz de ser autónomo un mínimo de 6 meses de operación, tomando medidas cada hora.

Posteriormente, con todos los cálculos y experimentos realizados con el entrenador Arduino UNO, se hará propio utilizando únicamente el Atmega 328 - P, que será la estimación de consumo de potencia real, es decir, la del diseño del dispositivo definitivo. De esta manera, se muestra el consumo excesivo de la placa Arduino UNO. Ya con los cálculos del consumo del MCU se podrá estimar el número de baterías necesario para que esta funcione durante los 6 meses mínimos de operación.

1. Potencia consumida teórica (Arduino UNO)

Para la primera estimación del consumo del dispositivo de medida, se va a utilizar el entrenador Arduino UNO. Esta placa va a ser la encargada de realizar las comunicaciones con los sensores cuando sea necesario, procesar su información y, posteriormente, volcar dicha información a través del módulo SD a la tarjeta de datos. Además, tras esto, tendrá que quedarse en modo inactivo hasta que pase el tiempo establecido para tomar otra medida.

El consumo del sistema se puede estudiar en dos partes. Por un lado, hay un consumo elevado durante un periodo de tiempo corto, que se corresponde con el calentamiento del sistema, la toma de medidas y el volcado de los datos. A esta parte del ciclo del programa de ejecución se llamará “ciclo operativo”. Por el otro lado, hay un consumo más reducido, que tendrá un periodo de tiempo bastante mayor, propio a cuando el Arduino Uno está durmiendo y, en consecuencia, los otros componentes del dispositivo se encuentran apagados, pues los

interruptores lógicos estarán abiertos. A esta parte del programa de ejecución se le llamará “ciclo inactivo”.

Por lo tanto, el consumo de corriente en el ciclo operativo es:

$$I_{Trabajo} = I_{Ard.Trabajo} + I_{O_2} + I_{CO_2} + I_{CO} + I_{SD}$$

Es decir, el consumo total vendrá establecido por la corriente solicitada por el Arduino, los sensores y la tarjeta de memoria.

Para conocer el consumo teórico de todos los componentes hay que ir a sus hojas de datos:

$$I_{Ard.Trabajo} = 46mA$$

$$I_{SD} = 15mA$$

$$I_{CO_2} = 10mA$$

Para los consumos de los otros sensores habrá que sacarlos a través de la potencia consumida, la cual es el dato eléctrico que aparece en la hoja de datos. De esta manera obtenemos que:

$$I_{O_2} = \frac{P}{V} = \frac{3.5mW}{5V} = 0.7mA$$

$$I_{O_2} = \frac{P}{V} = \frac{3mW}{5V} = 0.6mA$$

Quedando la corriente consumida por el sistema durante el ciclo operativo:

$$I_{Trabajo} = I_{Ard.Trabajo} + I_{O_2} + I_{CO_2} + I_{CO} + I_{SD}$$

$$I_{Trabajo} = 46mA + 0.7mA + 0.6mA + 10mA + 15mA$$

$$I_{Trabajo} = 72.4mA$$

Por otra parte, cuando el sistema se encuentra en ciclo inactivo, el único consumo que hay es el del Arduino en modo *Power Down*:

$$I_{Reposo} = I_{Ard.Sleep} = 20mA$$

Conociendo la corriente solicitada por cada uno de los elementos del circuito en cada momento y realizando un ciclo de medida cada hora, en el cual el sistema dispondrá de 15 segundos para realizar las medidas y el resto del tiempo del ciclo permanecerá inactivo obtendremos una gráfica de consumo:

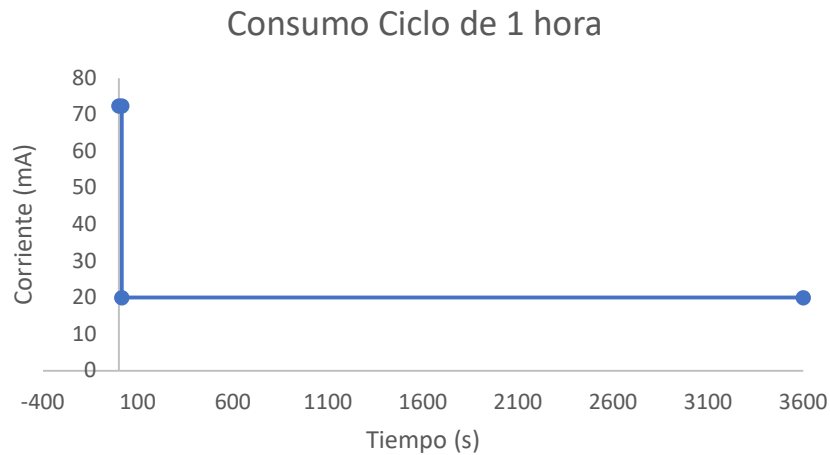


Figura 1: "Corriente teórica solicitada por el sistema durante ciclo de una hora".

En el gráfico se ha representado el consumo del teórico durante un ciclo completo. El pico de corriente se corresponde con el ciclo operativo, que dura alrededor de 11 segundos, y a continuación, una vez el sistema haya realizado todas las mediciones y el volcado de datos, una segunda zona de un consumo mucho menos elevado, que es el ciclo inactivo.

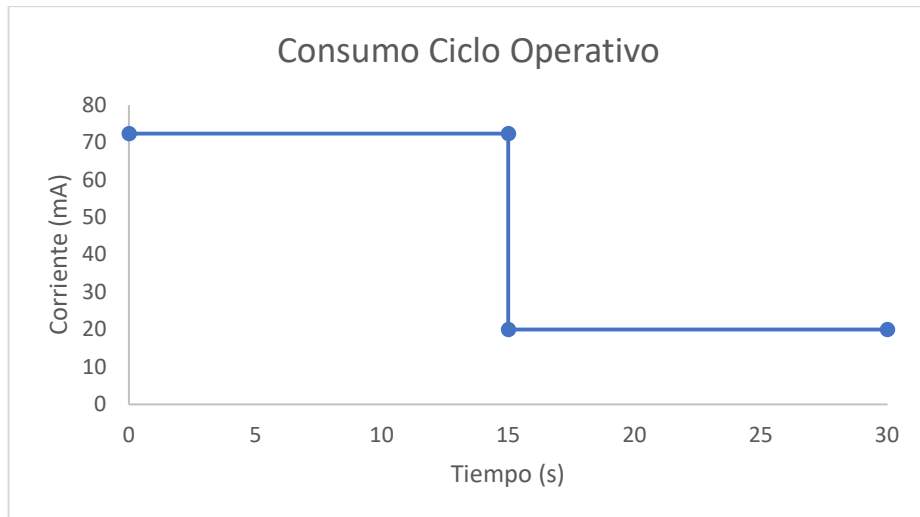


Figura 2: "Zoom del ciclo operativo teórico".

Este gráfico es un zoom del gráfico anterior, mostrando los primeros segundos del ciclo. Hay que recordar que son consumos teóricos, y por lo tanto las gráficas son ideales.

Para conocer cuál sería el consumo del sistema durante un ciclo completo hay que calcular la integral del área que encierra la función de la gráfica. Ya que esta gráfica es a tramos, es muy sencillo, ya que solo habría que calcular el área de ambos rectángulos y sumarlas. La energía consumida por el sistema quedaría tal que:

$$E_{Ciclo} = I_{Trabajo} * T_{Trabajo} + I_{Reposo} * T_{Reposo}$$

$$E_{Ciclo} = (I_{Ard.Trabajo} + I_{O_2} + I_{CO_2} + I_{CO} + I_{SD}) * T_{Trabajo} + I_{Ard.Sleep} * T_{Reposo}$$

$$E_{Ciclo} = (46mA + 0.7mA + 0.6mA + 10mA + 15mA) * \frac{15}{3600}h + 20mA * \frac{3585}{3600} * h$$

$$E_{Ciclo} = 20.22mAh$$

Con estos componentes y estableciendo una frecuencia de muestreo de una lectura cada hora obtenemos que el sistema consumirá 20.22 mAh por cada ciclo, es decir, cada hora.

2. Potencia consumida experimental (Arduino)

Una vez se ha estimado el consumo que tendría el sistema, se va a proceder a su montaje experimental, para hacer la medición del consumo real que tiene el sistema. De esta manera,

podremos verificar los cálculos realizados anteriormente. Además, con estas medidas se podrá hacer una estimación real de la duración de las baterías.

En primer lugar, hay que destacar que la placa Arduino UNO ha sido levemente modificado, para reducir así el consumo total de este primer prototipo, y asimilarlo lo más posible al consumo del sistema final. Esta reducción de consumo se ha hecho eliminando algunos elementos no funcionales existentes en la placa, tales como algunos leds indicadores. De esta manera, se ha hecho una primera reducción del consumo de la placa hasta los 36 mA, es decir, se han reducido 10 mA respecto al consumo teórico inicial.

Para la medición de la corriente solicitada por el sistema, se va a crear otro sistema que, recoja las medidas de corriente del sistema de medida en cada momento. Esto se va a realizar a través de otra placa Arduino UNO, la cual hará la función de amperímetro. Para la implementación de este sistema se va a colocar una resistencia de 1 ohm a la entrada del sistema, esto quiere decir que toda la corriente que el sistema solicite pasará por ella. De esta manera, se medirá con la placa Arduino la caída de tensión entre los bornes de la resistencia, dado que la resistencia tiene un valor de 1 ohm el valor de tensión medido en la resistencia será directamente la corriente solicitada por el sistema.

$$V = I * R$$

$$V = I * 1\Omega$$

Esta medición, debido a la resolución del Arduino, el cual para las entradas analógicas consta de un CAD de 10 bits, y con un rango de medición de las entradas analógicas de 0-5V tendrá una resolución de:

$$Resolución(V) = \frac{Rango}{2^n} = \frac{5V}{2^{10}} = 4.88mV$$

Con esta resolución del CAD, y conociendo la relación directa entre la tensión medida y la corriente solicitada, podemos saber que la resolución máxima a la hora de medir corriente será de:

$$Resolución(A) = \frac{Resolución(V)}{1\Omega} = 4.88mA$$

Esta resolución, debido a que las medidas que va a realizar el sistema, se encuentran entorno a unos 20-70 mA, es excesivamente alta lo que hará que no sea capaz de tomar las medidas de una forma exacta. Luego para mejorar la resolución se van a realizar una serie de modificaciones:

Por un lado, la placa Arduino es capaz de cambiar la referencia utilizada en el CAD. De esta manera, se puede cambiar esta referencia de su valor por defecto de 5V. En este caso, la mejor opción es el uso de una de las referencias internas del Arduino a 1.1 V. Así el rango de medidas de la entrada analógica es de 0 – 1.1 voltios.

El cambio se ha realizado mediante la función *analogReference()*, esta función viene ya implementada en el entorno de programación de Arduino IDE, por lo que solo hay que llamarla con el parámetro de entrada deseado, en este caso queremos que se coja la referencia interna de voltaje por lo que este cambio quedaría en el código tal que:

```
void setup()
{
    analogReference (INTERNAL);
}
```

Figura 3: "Comando utilizado para el cambio de referencia del CAD".

Quedando ahora la resolución de las entradas analógicas como:

$$Resolución(V) = \frac{Rango}{2^n} = \frac{1.1V}{2^{10}} = 1.07mV$$

Si volvemos a calcular la resolución en corriente quedaría tal que:

$$Resolución(A) = \frac{Resolución(V)}{1\Omega} = 1.07mA$$

Esta resolución es bastante más aceptable para las medidas que va a tener que tomar el sistema. No obstante, al realizar este cambio no se podrán tomar las medidas de forma directa en los bornes de la resistencia de 1 ohm, ya que en estos habrá una tensión entorno a 5V. Por esto, va a ser necesario la colocación de un partidor de tensión para adecuar la tensión de la fuente de alimentación a la tensión admisible por el CAD. Los valores de dicho partidor,

asumiendo que la tensión no va a sobrepasar los 5V, se va a optar por escalar la tensión a un sexto de su valor. Para esto, se han realizado los siguientes cálculos sobre el partidor de tensión:

$$V_o = V * \frac{R_2}{R_2 + R_1}$$

Si se desea que la tensión se escale a un sexto:

$$V_o = V * \frac{R_2}{R_2 + R_1} = V * \frac{1}{6}$$

$$6 * R_2 = R_2 + R_1$$

$$R_1 = 5 * R_2$$

Con este cálculo se ha optado por los valores de resistencia:

$$R_2 = 2k\Omega; R_1 = 5 * 2k\Omega = 10k\Omega$$

Gracias a estas dos modificaciones, por una parte, el cambio de referencia del CAD, aumentado su resolución, y por otra, adecuando la señal de medida, se ha obtenido un sistema capaz de medir la corriente solicitada por el sistema con una resolución de 1.07mA.

Ahora, con este sistema se va a proceder a su conexión y la recogida de sus medidas durante un ciclo completo de medida.

Esquema de conexión:

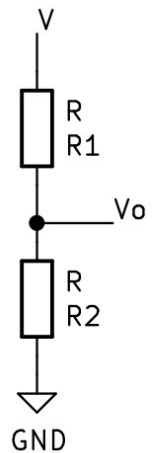


Figura 4: "Conexión del partidor".

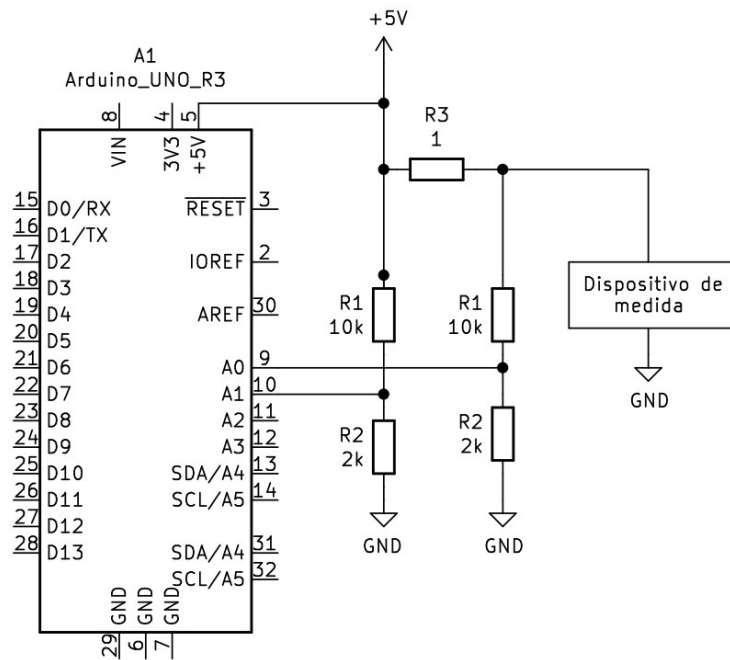


Figura 5: "Esquema general del sistema de medida".

Para tomar las medidas solo será necesaria la lectura de dos pines analógicos, uno en cada borne de la resistencia.

```

sensorValue = analogRead(sensorPin);
value = fmap(sensorValue, 0, 1023, 0.0, 1.1);
sensorValue2 = analogRead(sensorPin1);
value2 = fmap(sensorValue2, 0, 1023, 0.0, 1.1);

```

Figura 6: "Mapeado de las señales medidas al rango del CAD".

Las medidas recogidas son valores de 0-1024, ya que el CAD es de 10 bits, por esto es necesario cada vez que se toma la medida escalarla al rango de tensiones que queramos, en este caso ya que se ha usado la referencia interna se va a escalar a 1.1V.

Ya con estas medidas se ha procedido al cálculo de la caída de tensión y la corriente solicitada por el sistema.

```

value= value*6 ;
value2= value2*6;
Serial.print("Entrada: ");
Serial.println((value2),3);
Serial.print("Salida: ");
Serial.println(value,3);
Serial.print("Caida de tensión: ");
Serial.println(((value2-value)),3);
Serial.print("Corriente solicitada: ");
Serial.println((value2-value)/1.4,3);

```

Figura 7: “Código utilizado para el procesado de las medias recogidas”.

Por lo tanto, la corriente solicitada es la resta de los valores obtenidos por los bornes de la resistencia entre la resistencia de 1 ohmio. Sin embargo, la resistencia realmente tenía un valor de 1,4 ohmios, esa es la razón por la que se divide en el código entre 1.4.

De esta forma se puede realizar un muestreo del consumo del dispositivo. Tras dejar el sistema midiendo durante un ciclo de medida completo, que tiene una duración establecida de una hora, se obtiene:

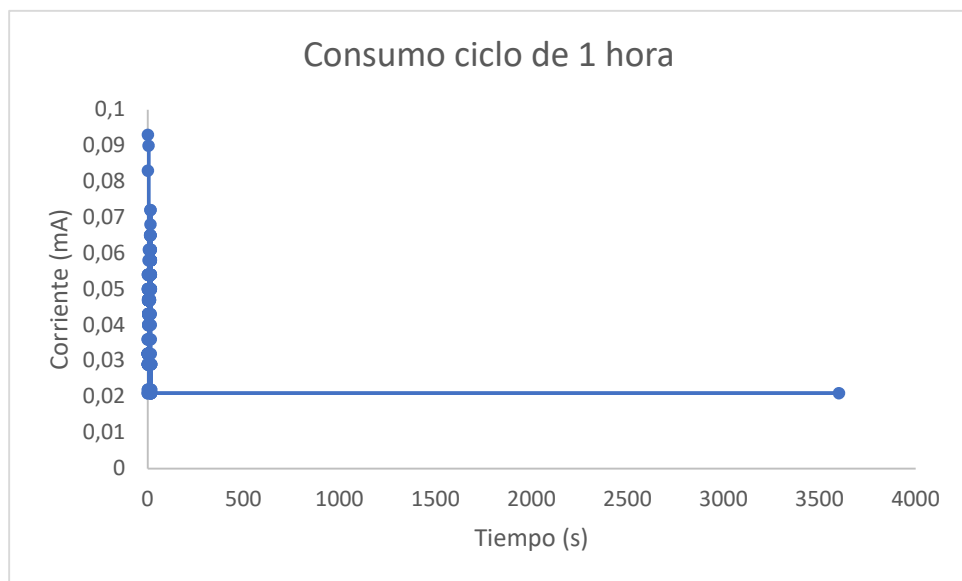


Figura 8: “Corriente experimental solicitada con Arduino UNO por el sistema, en un ciclo de una hora”.

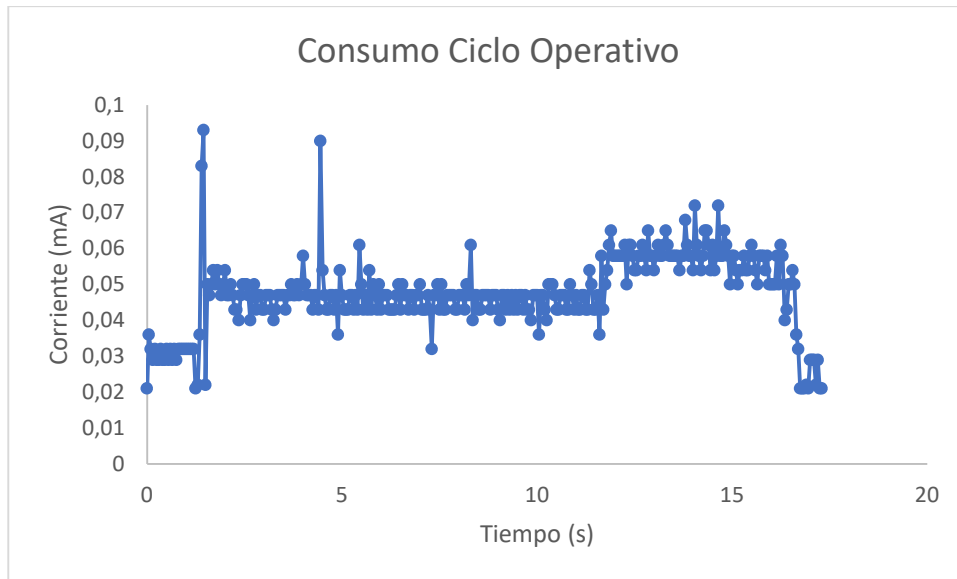


Figura 9: "Zoom del ciclo operativo experimental"

Como se puede ver en las gráficas, la función es similar a la gráfica con los consumos teóricos, con la diferencia de saltos en cada muestreo en el ciclo operativo, pues en la realidad el consumo varía dependiendo de la ejecución del programa

Con esta información se puede conocer la energía consumida por el sistema hallando el área bajo esta curva. Esta integración se va a realizar a través de la regla de los trapecios como en el apartado anterior. Estos cálculos se van a apoyar en la herramienta *Microsoft Excel*.

Tras el cálculo se ha obtenido que:

$$E_{Ciclo} = 21.13mA$$

Este consumo es bastante similar al que se ha realizado de forma teórica, por lo que se puede afirmar que este será el consumo del dispositivo si la placa se diseña con el Arduino UNO.

3. Estimación de tiempos (Arduino)

Una vez que se han hecho todos los cálculos, con sus debidas justificaciones teóricas, se ha procedido al cálculo del número de baterías que serán necesarias para que el sistema sea capaz de subsistir como mínimo 6 meses.

Estos cálculos se hacen partiendo del resultado práctico del consumo del sistema. Con este consumo por ciclo se puede estimar la energía necesaria para el suministro del sistema durante el tiempo deseado. Primero que nada, se ha calculado la energía consumida por el sistema en un periodo de un mes, tomando medidas cada hora.

$$E_{Mes} = E_{Ciclo} * \frac{24h}{1 dia} * \frac{30 dias}{1 mes}$$

$$E_{Mes} = 21.13mA * \frac{24h}{1 dia} * \frac{30 dias}{1 mes}$$

$$E_{Mes} = 15.21Ah$$

Con este dato, y conociendo la capacidad de las baterías de las que disponemos, se puede calcular de forma sencilla la energía teórica necesaria para el periodo de tiempo establecido:

$$N_{Baterias} = E_{Mes} * \frac{N_{Mes}}{E_{Bateria}}$$

$$N_{Baterias} = 15.21Ah * \frac{6 meses}{12 Ah}$$

$$N_{Baterias} = 7.6 Baterias$$

Es decir, para que el sistema sea capaz de permanecer 6 meses realizando medidas, habrá que colocar 8 baterías. Este número de baterías es asumible, ya que estas no son excesivamente grandes. No obstante, y como ya hemos comprobado, gran parte de toda esta energía se consumirá en componentes innecesarios para este proyecto de la placa Arduino. Por esta razón, se va a desechar la idea de prototipar el dispositivo con la placa Arduino y, en contraposición, sí hacerlo con el Atmega 328 – P disponiendo únicamente de los componentes necesarios.

4. **Potencia consumida teórica (Atmega328P)**

Para una mejora del rendimiento energético del sistema, se va a pasar a calcular el consumo del sistema, si se reemplaza la placa Arduino UNO, por el Atmega 328-P y los componentes necesarios para su funcionamiento.

El esquema es análogo al anterior, con la diferencia del cambio de componentes. A primera instancia, el consumo por parte del microcontrolador ya es notablemente inferior,

pasando de solicitar 46 mA en el ciclo operativo a 9.2 mA. Sin embargo, la parte más significativa de la reducción del consumo se da en el ciclo inactivo, que es el modo en que el dispositivo está durante más tiempo. En este caso, el consumo en reposo bajaría de 20 mA a tan solo 66 uA.

Estos cambios tan significativos reducen el consumo de energía por ciclo muy considerablemente:

$$E_{Ciclo} = I_{Trabajo} * T_{Trabajo} + I_{Reposo} * T_{Reposo}$$

$$E_{Ciclo} = (I_{ATMEGA.Trabajo} + I_{O_2} + I_{CO_2} + I_{CO} + I_{SD}) * T_{Trabajo} + I_{ATMEGA.Sleep} * T_{Reposo}$$

$$E_{Ciclo} = (9.2mA + 0.7mA + 0.6mA + 10mA + 15mA) * \frac{15}{3600} h + 0.066mA * \frac{3585}{3600} * h$$

$$E_{Ciclo} = 0.21 mAh$$

Este cambio, supone una reducción del consumo por ciclo del 99%. Esto hará que la autonomía del sistema aumente de manera sobresaliente.

5. Estimación de tiempos (Atmega 328P)

Con el cambio de microcontrolador y su consiguiente reducción del consumo general del sistema, se va a pasar a recalcular tanto la energía necesaria para los 6 meses, como el número de baterías necesarias para su autonomía.

Al igual que en cálculos anteriores se va a pasar a calcular primero la energía consumida por el sistema en un mes completo.

$$E_{Mes} = E_{Ciclo} * \frac{24h}{1 dia} * \frac{30 dias}{1 mes}$$

$$E_{Mes} = 0.21mA * \frac{24h}{1 dia} * \frac{30 dias}{1 mes}$$

$$E_{Mes} = 0.15Ah$$

Ya con este dato, y conociendo la capacidad de las baterías se va a volver a calcular el número necesario de estas de la misma manera que anteriormente.

$$N_{Baterias} = E_{Mes} * \frac{N_{Mes}}{E_{Bateria}}$$

$$N_{Baterias} = 0.15 Ah * \frac{6 meses}{12 Ah}$$

$$N_{Baterias} = 0.074 Baterias$$

Luego cambiando la placa Arduino UNO, por una placa de circuito impresa, con tan solo los elementos necesarios para este proyecto, se ha conseguido que el sistema pase de necesitar 8 baterías para subsistir los 6 meses establecidos a tan sólo necesitar una. De hecho, el cambio es tan drástico, que con tan solo una batería y configurando el sistema para tomar medidas cada hora, se podría dejar el sistema funcionando de manera permanente durante aproximadamente 168175 horas. Es decir, más de 7000 días. De esta manera se ha conseguido que el dispositivo sea muy eficiente energéticamente. Se ha pasado de ser las baterías el limitante del sistema, a ser ahora los propios componentes por desgaste en el tiempo.

6. Consumo experimental del sistema con ATMEGA328P

Para contrastar los cálculos realizados, se ha procedido a la conexión del MCU, fuera de la placa de desarrollo Arduino UNO. El chip ha sido conectado con tan solo los componentes básicos necesarios para su funcionamiento. Una vez ha sido conectado el chip, se ha procedido a la conexión de todos los pines necesarios a los sensores y a la tarjeta SD, como se refleja en el "ANEXO I: PLANOS" N°P. 2, para de esta manera, realizar una estimación del consumo final muy aproximado al real.

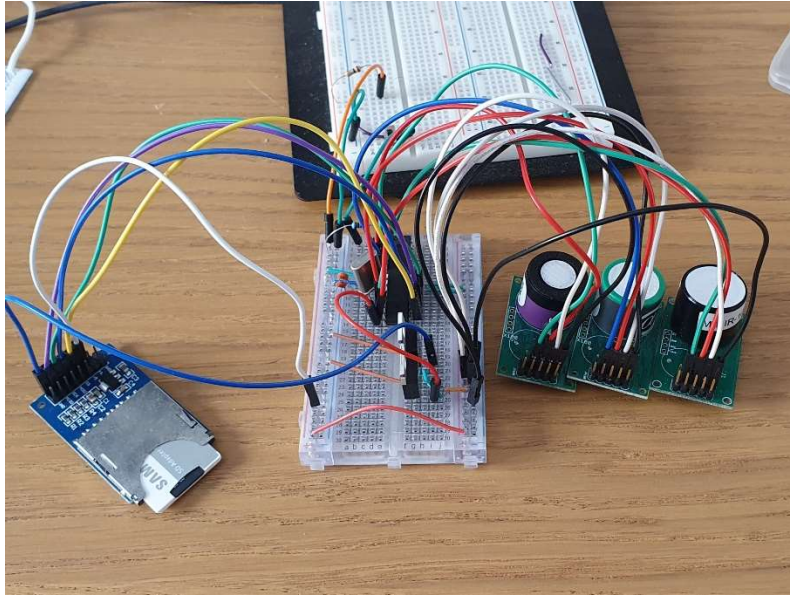
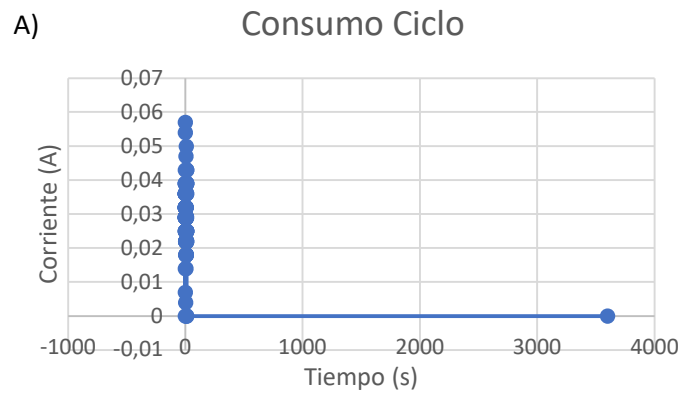


Figura 10: "Montaje experimental con ATMEGA 328P".

Se ha realizado la medición del consumo a través del mismo método utilizado anteriormente. Tras esto se han obtenido las siguientes gráficas de consumo:



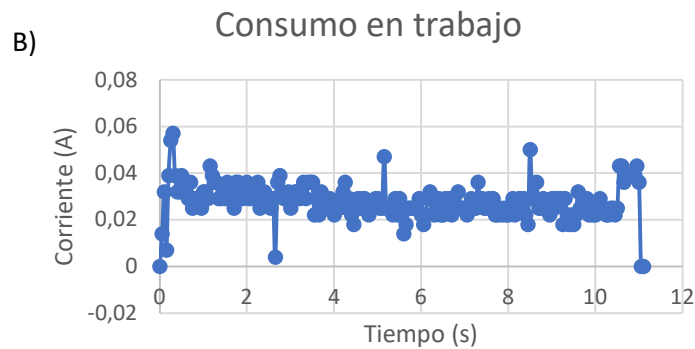


Figura 11: "A) Corriente solicitada experimental con ATMEGA328 P

B) Zoom del ciclo operativo"

Se puede apreciar como las gráficas siguen manteniendo el mismo formato, hasta ahora presente en las gráficas de consumo realizadas.

Debido a la resolución a la hora de tomar medidas de la que se dispone, ha sido imposible medir la corriente consumida por el chip cuando se encontraba en reposo, debido a su muy bajo valor. No obstante, este montaje ha servido para confirmar el bajo consumo del sistema una vez se haya imprimido su circuito. No obstante, para la estimación del cálculo de baterías, se ha asumido un consumo de 1mA cuando el sistema esté en reposo, el cual se corresponde a la resolución mínima de la que se dispone. Haciendo esto, y sabiendo que, en reposo, el sistema nunca sobrepasará dicho valor, se estimará la potencia consumida con este valor.

Con esta información se puede extrapolar la energía consumida por el sistema de la misma manera que se ha realizado en ocasiones anteriores. Realizando este cálculo con la ayuda de nuevo de *Microsoft Excel*.

$$E_{ciclo} = 1.083 \text{ mAh}$$

Con este redondeo al alza, la energía consumida en el ciclo y el número de baterías necesario para el sistema quedarían como:

$$E_{Mes} = E_{ciclo} * \frac{24h}{1 \text{ dia}} * \frac{30 \text{ dias}}{1 \text{ mes}}$$

$$E_{Mes} = 1.083 \text{ mA} * \frac{24h}{1 \text{ dia}} * \frac{30 \text{ dias}}{1 \text{ mes}}$$

$$E_{Mes} = 0.78Ah$$

Ya con este dato, y conociendo la capacidad de las baterías se va a volver a calcular el número necesario de estas de la misma manera que anteriormente.

$$N_{Baterias} = E_{Mes} * \frac{N_{Mes}}{E_{Bateria}}$$

$$N_{Baterias} = 0.78 \text{ Ah} * \frac{6 \text{ meses}}{12 \text{ Ah}}$$

$$N_{Baterias} = 0.39 \text{ Baterias}$$

Es decir, conectando una sola batería, no solo el sistema sería capaz de subsistir durante los 6 meses requeridos por el diseño. Sino que sería capaz de llegar a duplicar este tiempo, llegando hasta más de un año de autonomía.

Como norma general, la energía consumida por cada ciclo de medida se puede estimar a partir de la siguiente ecuación:

$$E_{Ciclo} = E_{C.Operativo} + E_{C.inactivo}$$

Por lo tanto, sabiendo que la energía del ciclo operativo no cambia, se puede estimar aplicando la integral de manera análoga a los cálculos anteriores, que tiene un valor de 0.086 mAh.

A esta energía, hay que sumarla la consumida por el sistema cuando se encuentra en modo inactivo. Sabiendo que el ciclo operativo dura 11 segundos, esta se puede calcular como:

$$E_{C.inactivo} = \frac{(T_{Usuario}(min) * 60 \frac{seg}{min}) - 11 \text{ seg}}{3600 \frac{seg}{h}} * 1mA$$

Quedando la energía de ciclo genérico como:

$$E_{Ciclo} = 0.086 \text{ mAh} + \frac{(T_{Usuario}(\text{min}) * 60 \frac{\text{seg}}{\text{min}}) - 11 \text{ seg}}{3600 \frac{\text{seg}}{\text{h}}} * 1 \text{ mA}$$

De esta manera, el usuario puede estimar la duración de las baterías con cualquier tiempo de muestreo que establezca.

ANEXO III

CÓDIGO

```

1
2 #include <SoftwareSerial.h>
3 #include "LowPower.h"
4 #include <SD.h>
5
6
7 //Declaramos qué pines vamos a utilizar como UART virtual
8
9 SoftwareSerial O2Serial(2, 3); // RX , TX OXIGENO
10 SoftwareSerial CO2Serial(6, 7); // RX , TX CO2
11 SoftwareSerial COSerial(4, 5); // RX , TX CO
12
13 //-----DECLARACION DE TIEMPO ENTRE MEDIDAS-----
14 /*INTRODUZCA EN LA VARIABLE TIEMPO EL TIEMPO QUE DESEE ENTRE MEDIDAD.
15 Ej: TIEMPO = 15; TOMA DE MEDIDAS CADA 15 MIN*/
16 int Tiempo = 60;
17 //-----DECLARACIÓN DE NÚMERO DE MEDIDAS POR FICHERO-----
18 /*INTRODUZCA EN LA VARIABLE M_FICH EL NÚMERO DE MEDIDAS QUE DESEE QUE CONTENGA
19 CADA FICHERO Ej: M_FICH = 24; SE GUARDARÁN EN CADA FICHERO 24 MEDIDAS.
20 SI SE COMBINA ESTO CON UNA FRECUENCIA DE MUESTREO CADA HORA, SE DISPONDRÁ
21 DE UN FICEHRO DIARIO*/
22 int N_FICH = 24;
23 //-----DECLARACIÓN DE VARIABLES-----
24 String codigo;
25 String datos;
26 File myFile;
27 int N_Semana;
28 int contador;
29 String nombreFich;
30 //-----INICIALIZACIÓN DE FUNCIONES-----
31 void date();
32 void leer();
33 void temp();
34 void pressure();
35 void Humidity();
36 void concentrationO2();
37 void concentrationCO2();
38 void concentrationCO();
39 void sleep();
40 void SDCARD();
41 void leerCO2();
42 void selNombre();
43 const int MOSFET = 8;
44
45 //-----INICIALIZACIÓN DEL SISTEMA-----
46 void setup()
47 {
48 //Apertura de las comunicaciones seriales e inicialización del baud rate
49 Serial.begin(9600);
50 O2Serial.begin(9600);
51 CO2Serial.begin(9600);
52 COSerial.begin(9600);
53 // Declaración de los pines CS y MOSFET como salidas
54 pinMode(10,OUTPUT);
55 pinMode(MOSFET,OUTPUT);
56
57 //-----CHEQUEO DE LOS PUERTOS SERIALES CREADOS-----
58 while (!O2Serial) {
59
60 }
61 while (!CO2Serial) {
62
63 }
64 while (!COSerial) {
65
66 }
67
68 // ----- INICIALIZACIÓN DE LA TARJETA SD-----
69 digitalWrite(MOSFET , HIGH);
70 if (!SD.begin(10)) {
71 Serial.println("initialization failed!");
72 digitalWrite(MOSFET , LOW);
73 while (1);

```

```

74     }
75     //-----DEFINICIÓN DEL PRIMER NOMBRE DEJ FICHERO-----
76     N_Semana = 1;
77     contador = 0;
78     nombreFich = "Fich" + String(N_Semana) + ".csv";
79
80     //-----SE ELIMINA FICHEROS CREADOS ANTES DE LA PRIMERA MEDIDA-----
81     if (SD.exists(nombreFich)) {
82         Serial.println(F("El fichero ya existia"));
83         SD.remove(nombreFich);
84         Serial.println(F("Fichero eliminado"));
85     } else {
86         Serial.println("Fichero1.csv no existia");
87     }
88     //-----CREACIÓN DE FICHERO Y ESCRITURA DE ENCABEZADO-----
89     myFile = SD.open(nombreFich, FILE_WRITE);        //APERTURA DEL FICHERO CON
90     PERMISO DE ESCRITURA Y LECTURA
91     if (myFile) {
92         Serial.println("Archivo nuevo, Escribiendo encabezado");
93         myFile.println("Fecha, Hora, Temperatura (°C), Presion (mbar), Humedad
94         (%), O2 (*10^3 ppm), CO2 (ppm), CO (ppm)");
95         myFile.close();
96     } else {
97         Serial.println("Error creando el archivo datos.csx");
98     }
99     Serial.println(F("TRABAJO FIN DE GRADO"));
100    Serial.println(F("Fecha: 07/04/2020"));
101    Serial.println(F("Autores: Alberto Doro García"));
102    Serial.println(F("Ezequiel Cortés Martín\n"));
103    Serial.println(F("----- PROGRAMA: PRUEBA Y VALIDACIÓN DE MEDIDAS -----"));
104    Serial.println(F("El sistema se está calentado."));
105    Serial.println(F("Rl sistema ya está listo para medir. \n"));
106    Serial.println(F("Fecha, Hora, Temperatura (°C), Presion (mbar), Humedad (%), O2 (*10^3
107    ppm), CO2 (ppm), CO (ppm)"));
108 }
109 void loop() // run over and over
110 {
111     //-----ENCENDIDO Y CALENTADO DEL SISTEMA-----
112     digitalWrite(MOSFET , HIGH);    // MOSFET EN SATURACIÓN
113     delay(10000);
114     //-----COMPROBAR INICIALIZACIÓN DE LA TARJETA SD-----
115     if (!SD.begin(10)) {
116         Serial.println("initialization failed!");
117         digitalWrite(MOSFET , LOW);
118         while (1);
119         //SI FALLARA LA INICIALIZACIÓN EL SISTEMA SE BLOQUEARÁ AQUI
120         //PARA EVITAR POSIBLES DAÑOS A LOS EQUIPOS
121     }
122     date();          //COMUNICACIÓN CON LOS SENORES PARA LA OBTENCIÓN DE FECHA Y
123     HORA
124     temp();          //COMUNICACIÓN CON LOS SENORES PARA LA OBTENCIÓN DE
125     TEMPERATURA
126     pressure();      //COMUNICACIÓN CON LOS SENORES PARA LA OBTENCIÓN DE PRESIÓN
127     Humidity();      //COMUNICACIÓN CON LOS SENORES PARA LA OBTENCIÓN DE HUMEDAD
128     concentrationO2(); //COMUNICACIÓN CON EL SENOR DE O2 PARA LA OBTENCIÓN DE SU
129     CONCENTRACIÓN
130     concentrationCO2(); //COMUNICACIÓN CON EL SENOR DE CO2 PARA LA OBTENCIÓN DE SU
131     CONCENTRACIÓN
132     concentrationCO(); //COMUNICACIÓN CON EL SENOR DE CO PARA LA OBTENCIÓN DE SU
133     CONCENTRACIÓN
134     selNombre();     //COMPROBACIÓN DEL NOMBRE DEL FICHERO
135     delay(30);
136     SDCARD();        //VOLCAR DATOS A LA TARJETA SD
137     delay(50);
138     digitalWrite(MOSFET , LOW);    //MOSFET EN CORTE
139     delay(100);
140     sleep();         //INTRODUCCIÓN DEL MCU EN SU ESTADO DE POWER-DOWN
141     delay(50);
142 }

```

```

139  /*FUNCIÓN PARA LA LECTURA DE INFORMACIÓN A TRAVES DE LA UART DEL SENSOR DE O2 */
140
141  void leerO2 ()
142  {
143      //ESTA FUNCIÓN LEE EL BUFFER DE LA UART DE O2 HASTA QUE ESTE SE VACÍA
144      while (O2Serial.available() > 0)
145      {
146          char inByte = O2Serial.read();
147          codigo += inByte;
148      }
149
150  }
151  //FUNCIÓN PARA LA LECTURA DE INFORMACIÓN A TRAVES DE LA UART DEL SENSOR DE CO2
152  void leerCO2 ()
153  {
154      //ESTA FUNCIÓN LEE EL BUFFER DE LA UART DE CO2 HASTA QUE ESTE SE VACÍA
155      while (CO2Serial.available() > 0)
156      {
157          char inByte = CO2Serial.read();
158          codigo += inByte;
159      }
160
161  }
162  //FUNCIÓN PARA LA LECTURA DE INFORMACIÓN A TRAVES DE LA UART DEL SENSOR DE CO
163  void leerCO ()
164  {
165      //ESTA FUNCIÓN LEE EL BUFFER DE LA UART DE CO HASTA QUE ESTE SE VACÍA
166      while (COSerial.available() > 0)
167      {
168          char inByte = COSerial.read();
169          codigo += inByte;
170      }
171
172  }
173
174  void date () {
175      /*COMUNICACIÓN CON LOS SENORES PARA LA OBTENCIÓN DE FECHA Y HORA
176      COMANDO DE FECHA Y HORA "c 13" */
177      O2Serial.println("c 13");
178      O2Serial.listen();
179      delay(20);
180      //LECTURA DEL BUFFER DEL SENSOR
181      leerO2 ();
182      //ORGANIZACIÓN DE LA INFORMACIÓN EN FORMATO CSV
183      String fecha = codigo.substring(1, 12) + "," + codigo.substring(13, 21) + ","; //
184      + " ";
185      datos += fecha;
186      codigo = ""; //vaciamos la variable
187  }
188
189  void temp () {
190      /*COMUNICACIÓN CON LOS SENORES PARA LA OBTENCIÓN DE FECHA Y HORA
191      COMANDO DE TEMPERATURA "T 13" */
192      O2Serial.println("T 13");
193      O2Serial.listen();
194      delay(20);
195      //LECTURA DEL BUFFER DEL SENSOR
196      leerO2 ();
197      //ORGANIZACIÓN DE LA INFORMACIÓN EN FORMATO CSV
198      String temperatura = codigo.substring(4, 6) + "." + codigo[6] + ",";
199      datos += temperatura;
200      codigo = "";
201  }
202
203  void pressure () {
204      //COMUNICACIÓN CON LOS SENORES PARA LA OBTENCIÓN DE FECHA Y HORA
205      COMANDO DE PRESIÓN "B 13" */
206      O2Serial.println("B 13");
207      O2Serial.listen();
208      delay(20);
209      //LECTURA DEL BUFFER DEL SENSOR
210      leerO2 ();
211      //ORGANIZACIÓN DE LA INFORMACIÓN EN FORMATO CSV

```

```

211     String presion = codigo.substring(2, (codigo.length()-3)) +
        codigo[codigo.length()-3] + ","; // + " ";
212     datos += presion;
213     codigo = "";
214 }
215
216 void Humidity(){
217     //COMUNICACIÓN CON LOS SENORES PARA LA OBTENCIÓN DE FECHA Y HORA
218     COMANDO DE HUMEDAD "H 13" */
219     O2Serial.println("H 13");
220     O2Serial.listen();
221     delay(20);
222     //LECTURA DEL BUFFER DEL SENSOR
223     leerO2();
224     //ORGANIZACIÓN DE LA INFORMACIÓN EN FORMATO CSV
225     String humedad = codigo.substring(2, (codigo.length()-3)) + "." +
        codigo[codigo.length()-3] + ","; // + " ";
226     datos += humedad;
227     codigo = "";
228 }
229
230 void concentrationO2(){
231     //COMUNICACIÓN CON EL SENOR DE O2 PARA LA OBTENCIÓN DE SU CONCENTRACIÓN
232     COMANDO DE HUMEDAD "Z 13" */
233     O2Serial.println("Z 13");
234     O2Serial.listen();
235     delay(20);
236     //LECTURA DEL BUFFER DEL SENSOR
237     leerO2();
238     //ORGANIZACIÓN DE LA INFORMACIÓN EN FORMATO CSV
239     String O2 = codigo.substring(2, (codigo.length()-4)) + "." +
        codigo.substring(codigo.length()-4, (codigo.length()-2)) + ","; // +
        " ";
240     datos += O2;
241     codigo = "";
242 }
243
244 void concentrationCO2(){ /* MIRAR LA FUNCION LEER*/
245     //COMUNICACIÓN CON EL SENOR DE CO2 PARA LA OBTENCIÓN DE SU CONCENTRACIÓN
246     COMANDO DE HUMEDAD "Z 13" */
247     CO2Serial.println("Z 13");
248     CO2Serial.listen();
249     delay(20);
250     //LECTURA DEL BUFFER DEL SENSOR
251     leerCO2();
252     //ORGANIZACIÓN DE LA INFORMACIÓN EN FORMATO CSV
253     String CO2 = codigo.substring(2, (codigo.length()-2)) + "00,";
254     datos += CO2;
255     codigo = "";
256 }
257
258 void concentrationCO(){
259     /*COMUNICACIÓN CON EL SENOR DE CO2 PARA LA OBTENCIÓN DE SU CONCENTRACIÓN
260     COMANDO DE HUMEDAD "Z 13" */
261     COSerial.println("Z 13");
262     COSerial.listen();
263     delay(20);
264     //LECTURA DEL BUFFER DEL SENSOR
265     leerCO();
266     //ORGANIZACIÓN DE LA INFORMACIÓN EN FORMATO CSV
267     String CO = codigo.substring(3, (codigo.length()-2));
268     datos += CO;
269     codigo = "";
270 }
271
272 void sleep(){
273     /* FUNCIÓN DE MODO POWER-DOWN:
274     SE HA REALIZADO EL CÁLCULO DEL NÚMERO DE ITERACIONES
275     NECESARIAS PARA ALCANZAR EL PERIODO DE REPOSO ESPECIFICADO.
276     BASTA CON LA CONVERSIÓN DEL TIEMPO INTRODUCIDO POR EL USUARIO
277     A SEGUNDOS Y POSTERIORMENTE LA DIVISIÓN POR EL PERIODO MÁXIMO DE 8 SEGUNDOS DEL
278     TEMPORIZADOR DEL WATCHDOG */
279     delay(20);

```

```

280     int T = round(((Tiempo*60)-Tiempo*0.6)/8.6);
281     Serial.println(T);
282     delay(20);
283     //REALIZACIÓN DE LAS ITERACIONES CALCULADAS ANTERIORMENTE
284     for(int i = 0; i < T ; i++){
285         LowPower.powerDown(SLEEP_8S, ADC_OFF, BOD_OFF);
286     }
287     delay(20);
288 }
289
290 void SDCARD() {
291     Serial.println(datos);
292     //APERTURA DEL FICHERO CON PERMISO DE ESCRITURA Y LECTURA
293     myFile = SD.open(nombreFich, FILE_WRITE);
294     if (myFile) {
295         Serial.println("El archivo se ha abierto y está escribiendo...");
296         //VOLCADO DE DATOS RECOGIDOS POR LOS SENSORES YA PROCESADOS
297         myFile.println(datos);
298         myFile.close();
299     } else {
300
301         Serial.println("Error volcando datos");
302     }
303     //CIERRE DEL FICHERO
304     myFile = SD.open(nombreFich);
305
306     /*
307     if (myFile){
308         Serial.println(nombreFich);
309         while (myFile.available()) {
310             Serial.write(myFile.read());
311         }
312         myFile.close();
313
314     } else {
315         Serial.println(F("No se pudo mostrar el contenido del fichero"));
316     }*/
317     datos = "";
318
319 }
320 void selNombre(){
321     /*FUNCIÓN DE CONTROL DE NOMNRE DE LOS ARCHIVOS:
322     ESTA FUNCIÓN COMPORBARÁ CUANTAS LINEAS DE DATOS SE HAN INTRODUCIDO
323     EN UN FICHERO, E INCREMENTARÁ SU NÚMERO DE FICHERO CUANDO SE
324     ALCANCE EL NÚMERO DE DATOS POR FICHERO ESPECIFICADOS POR EL USUARIO
325     EN N_FICH*/
326     contador++;
327     if(contador > N_FICH){
328         //CUANDO SE ALCANCE DICHO NÚMERO DE LINEAS DE DATOS N_FICH
329         //SE INCREMENTARÁ EL CONTADOR Y SE CAMBIARÁ EL NOMBRE DEL FICHERO
330         N_Semana++;
331         cõtador = 1;
332         nombreFich = "Fich" + String(N_Semana) + ".csv";
333         //CADA VEZ QUE SE CAMBIE EL NOMBRE DEL FICHERO SE COMPROBARÁ
334         //QUE NO EXISTEN ARCHIVOS DUPLICADOS, ELIMINANDO ESTOS SI EXISTIERAN
335         if (SD.exists(nombreFich)) {
336             SD.remove(nombreFich);
337         }
338     }
339     /*APERTURA DEL FICHERO CON PERMISO DE ESCRITURA Y LECTURA
340     PARA LA COMPROBACIÓN DE SU CORRECTA CREACIÓN*/
341     myFile = SD.open(nombreFich, FILE_WRITE);
342     if (!myFile) {
343         Serial.println("Error creando el archivo datos.csv");
344         while(1);
345     }
346     myFile.close();
347 }
348

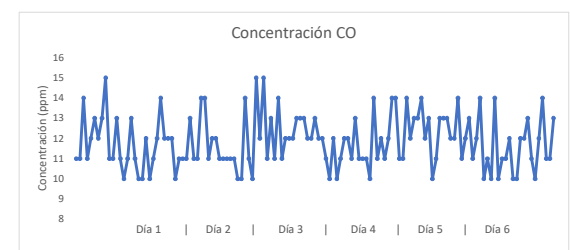
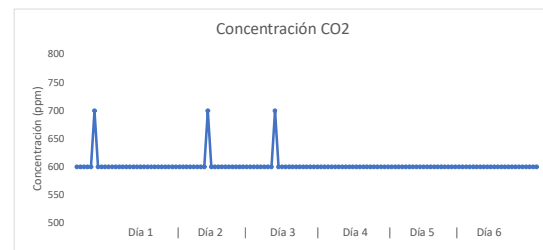
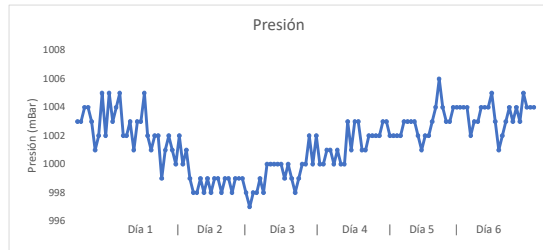
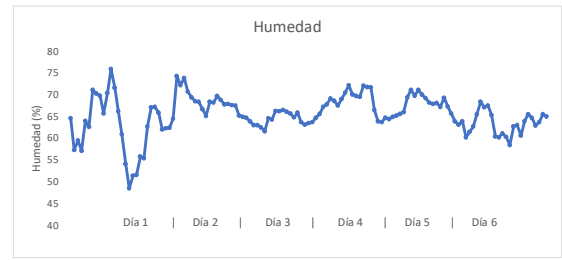
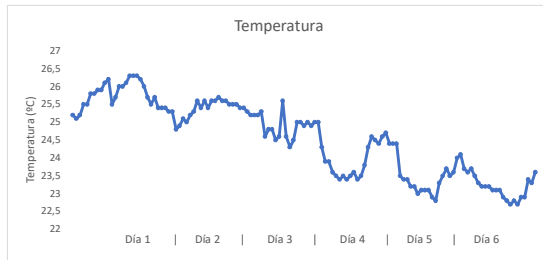
```

ANEXO IV

MEDIDAS

Número	Fichero	Fecha	Hora	Temperatura (°C)	Presión (mbar)	Humedad (%)	Concentración O2	Concentración CO2 (ppm)	Concentración CO (ppm)
1	FICH2.CSV	17/06/2020	6:21:35	25,2	1003	64,7	20104	600	11
2	FICH3.CSV	17/06/2020	7:20:16	25,1	1003	57,4	20120	600	11
3	FICH4.CSV	17/06/2020	8:18:57	25,2	1004	59,6	20112	600	14
4	FICH5.CSV	17/06/2020	9:17:38	25,5	1004	57,2	20146	600	11
5	FICH6.CSV	17/06/2020	10:16:20	25,5	1003	64,1	20120	600	12
6	FICH7.CSV	17/06/2020	11:15:01	25,8	1001	62,7	20134	700	13
7	FICH8.CSV	17/06/2020	12:13:44	25,8	1002	71,2	20110	600	12
8	FICH9.CSV	17/06/2020	13:12:26	25,9	1005	70,4	20108	600	13
9	FICH10.CSV	17/06/2020	14:11:08	25,9	1002	69,9	20116	600	15
10	FICH11.CSV	17/06/2020	15:09:50	26,1	1005	65,8	20143	600	11
11	FICH12.CSV	17/06/2020	16:08:33	26,2	1003	70,5	20125	600	11
12	FICH13.CSV	17/06/2020	17:07:16	25,5	1004	76	20094	600	13
13	FICH14.CSV	17/06/2020	18:05:57	25,7	1005	71,7	20104	600	11
14	FICH15.CSV	17/06/2020	19:04:39	26	1002	66,3	20135	600	10
15	FICH16.CSV	17/06/2020	20:03:21	26	1002	61	20149	600	11
16	FICH17.CSV	17/06/2020	21:02:04	26,1	1003	54,2	20170	600	13
17	FICH18.CSV	17/06/2020	22:00:47	26,3	1001	48,6	20200	600	11
18	FICH19.CSV	17/06/2020	22:59:29	26,3	1003	51,5	20200	600	10
19	FICH20.CSV	17/06/2020	23:58:13	26,3	1003	51,7	20192	600	10
20	FICH21.CSV	18/06/2020	0:56:56	26,2	1005	55,9	20174	600	12
21	FICH22.CSV	18/06/2020	1:55:38	26	1002	55,5	20172	600	10
22	FICH23.CSV	18/06/2020	2:54:21	25,7	1001	62,8	20132	600	11
23	FICH24.CSV	18/06/2020	3:53:01	25,5	1002	67,2	20106	600	12
24	FICH25.CSV	18/06/2020	4:51:42	25,7	1002	67,3	20108	600	14
25	FICH26.CSV	18/06/2020	5:50:24	25,4	999	66	20104	600	12
26	FICH27.CSV	18/06/2020	6:49:05	25,4	1001	62,1	20114	600	12
27	FICH28.CSV	18/06/2020	7:47:46	25,4	1002	62,4	20112	600	12
28	FICH29.CSV	18/06/2020	8:46:27	25,3	1001	62,5	20112	600	10
29	FICH30.CSV	18/06/2020	9:45:08	25,3	1000	64,6	20108	600	11
30	FICH31.CSV	18/06/2020	10:43:48	24,8	1002	74,4	20079	600	11
31	FICH32.CSV	18/06/2020	11:42:28	24,9	1000	72,3	20065	600	11
32	FICH33.CSV	18/06/2020	12:41:08	25,1	1001	73,9	20065	600	13
33	FICH34.CSV	18/06/2020	13:39:47	25	999	70,8	20081	600	11
34	FICH35.CSV	18/06/2020	14:38:27	25,2	998	69,5	20085	600	11
35	FICH36.CSV	18/06/2020	15:37:07	25,3	998	68,6	20087	600	14
36	FICH37.CSV	18/06/2020	16:35:48	25,6	999	68,5	20100	600	14
37	FICH38.CSV	18/06/2020	17:34:29	25,4	998	66,8	20104	600	11
38	FICH39.CSV	18/06/2020	18:33:10	25,6	999	65,2	20106	700	12
39	FICH40.CSV	18/06/2020	19:31:51	25,4	998	68,5	20094	600	12
40	FICH41.CSV	18/06/2020	20:30:32	25,6	999	68,3	20100	600	11
41	FICH42.CSV	18/06/2020	21:29:13	25,6	999	69,8	20098	600	11
42	FICH43.CSV	18/06/2020	22:27:54	25,7	998	68,9	20108	600	11
43	FICH44.CSV	18/06/2020	23:26:35	25,6	999	67,9	20110	600	11
44	FICH45.CSV	19/06/2020	0:25:16	25,6	999	68	20104	600	11
45	FICH46.CSV	19/06/2020	1:23:57	25,5	998	67,7	20110	600	10
46	FICH47.CSV	19/06/2020	2:22:38	25,5	999	67,6	20104	600	10
47	FICH48.CSV	19/06/2020	3:21:19	25,5	999	65,3	20118	600	14
48	FICH49.CSV	19/06/2020	4:20:00	25,4	999	65	20114	600	11
49	FICH50.CSV	19/06/2020	5:18:40	25,4	998	64,8	20108	600	10
50	FICH51.CSV	19/06/2020	6:17:21	25,3	997	64	20108	600	15
51	FICH52.CSV	19/06/2020	7:16:02	25,2	998	63,1	20108	600	12
52	FICH53.CSV	19/06/2020	8:14:42	25,2	998	63,1	20106	600	15
53	FICH54.CSV	19/06/2020	9:13:23	25,2	999	62,6	20112	600	11
54	FICH55.CSV	19/06/2020	10:12:03	25,3	998	61,7	20120	600	13
55	FICH56.CSV	19/06/2020	11:10:44	24,6	1000	64,7	20112	600	11
56	FICH57.CSV	19/06/2020	12:09:23	24,8	1000	64,4	20106	600	14
57	FICH58.CSV	19/06/2020	13:08:03	24,8	1000	66,4	20075	700	11
58	FICH59.CSV	19/06/2020	14:06:42	24,5	1000	66,3	20095	600	12
59	FICH60.CSV	19/06/2020	15:05:21	24,6	1000	66,6	20089	600	12
60	FICH61.CSV	19/06/2020	16:04:01	25,6	999	66,2	20106	600	12
61	FICH62.CSV	19/06/2020	17:02:42	24,6	1000	65,8	20114	600	13
62	FICH63.CSV	19/06/2020	18:01:21	24,3	999	64,9	20098	600	13
63	FICH64.CSV	19/06/2020	19:00:00	24,5	998	66	20075	600	13
64	FICH65.CSV	19/06/2020	19:58:39	25	999	63,8	20094	600	12
65	FICH66.CSV	19/06/2020	20:57:19	25	1000	63,2	20102	600	12
66	FICH67.CSV	19/06/2020	21:55:59	24,9	1000	63,6	20100	600	13
67	FICH68.CSV	19/06/2020	22:54:39	25	1002	63,8	20098	600	12
68	FICH69.CSV	19/06/2020	23:53:19	24,9	1000	64,8	20098	600	12
69	FICH70.CSV	20/06/2020	0:51:59	25	1002	65,7	20094	600	11
70	FICH71.CSV	20/06/2020	1:50:39	25	1000	67,3	20085	600	10
71	FICH72.CSV	20/06/2020	2:49:18	24,3	1000	67,9	20109	600	12
72	FICH73.CSV	20/06/2020	3:47:56	23,9	1001	69,2	20093	600	10
73	FICH74.CSV	20/06/2020	4:46:34	23,9	1001	68,7	20083	600	11
74	FICH75.CSV	20/06/2020	5:45:11	23,6	1000	67,6	20103	600	12
75	FICH76.CSV	20/06/2020	6:43:48	23,5	1001	69,1	20087	600	12
76	FICH77.CSV	20/06/2020	7:42:25	23,4	1000	70,6	20086	600	11
77	FICH78.CSV	20/06/2020	8:41:02	23,5	1000	72,3	20062	600	13
78	FICH79.CSV	20/06/2020	9:39:38	23,4	1003	70,2	20082	600	11
79	FICH80.CSV	20/06/2020	10:38:15	23,5	1001	69,8	20082	600	11
80	FICH81.CSV	20/06/2020	11:36:52	23,6	1003	69,6	20085	600	11
81	FICH82.CSV	20/06/2020	12:35:29	23,4	1003	72,2	20086	600	10
82	FICH83.CSV	20/06/2020	13:34:06	23,5	1001	71,9	20080	600	14
83	FICH84.CSV	20/06/2020	14:32:43	23,8	1001	71,8	20077	600	11
84	FICH85.CSV	20/06/2020	15:31:21	24,3	1002	66,6	20089	600	12
85	FICH86.CSV	20/06/2020	16:29:59	24,6	1002	64	20097	600	11
86	FICH87.CSV	20/06/2020	17:28:38	24,5	1002	63,8	20097	600	12
87	FICH88.CSV	20/06/2020	18:27:17	24,4	1002	64,8	20097	600	14
88	FICH89.CSV	20/06/2020	19:25:56	24,6	1003	64,5	20093	600	14
89	FICH90.CSV	20/06/2020	20:24:36	24,7	1003	65	20091	600	11
90	FICH91.CSV	20/06/2020	21:23:15	24,4	1002	65,3	20103	600	11
91	FICH92.CSV	20/06/2020	22:21:53	24,4	1002	65,7	20099	600	14
92	FICH93.CSV	20/06/2020	23:20:32	24,4	1002	66,1	20097	600	12
93	FICH94.CSV	21/06/2020	0:19:10	23,5	1002	69,5	20097	600	13
94	FICH95.CSV	21/06/2020	1:17:47	23,4	1003	71,2	20086	600	13
95	FICH96.CSV	21/06/2020	2:16:24	23,4	1003	69,9	20088	600	14
96	FICH97.CSV	21/06/2020	3:15:01	23,2	1003	71,2	20086	600	12
97	FICH98.CSV	21/06/2020	4:13:38	23,2	1003	70,1	20088	600	13
98	FICH99.CSV	21/06/2020	5:12:14	23	1002	69,3	20092	600	10
99	FICH100.CSV	21/06/2020	6:10:50	23,1	1001	68,3	20090	600	11
100	FICH101.CSV	21/06/2020	7:09:26	23,1	1002	68	20090	600	13
101	FICH102.CSV	21/06/2020	8:08:03	23,1	1002	68,2	20084	600	13
102	FICH103.CSV	21/06/2020	9:06:39	22,9	1003	67,3	20094	600	13
103	FICH104.CSV	21/06/2020	10:05:15	22,8	1004	69,4	20082	600	12
104	FICH105.CSV	21/06/2020	11:03:51	23,3	1006	67,4	20084	600	12
105	FICH106.CSV	21/06/2020	12:02:28	23,5	1004	65,8	20101	600	14
106	FICH107.CSV	21/06/2020	13:01:05	23,7	1003	64	20101	600	11
107	FICH108.CSV	21/06/2020	13:59:43	23,5	1003	63,2	20107	600	12
108	FICH109.CSV	21/06/2020	14:58:20	23,6	1004	64	20099	600	13

109	FICH110.CSV	21/06/2020	15:56:58	24	1004	60,3	20115	600	11
110	FICH111.CSV	21/06/2020	16:55:36	24,1	1004	61,6	20093	600	12
111	FICH112.CSV	21/06/2020	17:54:13	23,7	1004	62,8	20105	600	14
112	FICH113.CSV	21/06/2020	18:52:50	23,6	1004	65,6	20110	600	10
113	FICH114.CSV	21/06/2020	19:51:27	23,7	1002	68,5	20099	600	11
114	FICH115.CSV	21/06/2020	20:50:04	23,5	1003	67,2	20097	600	10
115	FICH116.CSV	21/06/2020	21:48:41	23,3	1003	67,6	20090	600	14
116	FICH117.CSV	21/06/2020	22:47:17	23,2	1004	65,4	20107	600	10
117	FICH118.CSV	21/06/2020	23:45:53	23,2	1004	60,5	20111	600	11
118	FICH119.CSV	22/06/2020	0:44:30	23,2	1004	60,3	20119	600	11
119	FICH120.CSV	22/06/2020	1:43:06	23,1	1005	61,2	20115	600	12
120	FICH121.CSV	22/06/2020	2:41:42	23,1	1003	60,4	20117	600	10
121	FICH122.CSV	22/06/2020	3:40:18	23,1	1001	58,5	20119	600	10
122	FICH123.CSV	22/06/2020	4:38:54	22,9	1002	62,8	20117	600	12
123	FICH124.CSV	22/06/2020	5:37:30	22,8	1003	63,1	20104	600	12
124	FICH125.CSV	22/06/2020	6:36:06	22,7	1004	60,7	20121	600	13
125	FICH126.CSV	22/06/2020	7:34:42	22,8	1003	64	20102	600	11
126	FICH127.CSV	22/06/2020	8:33:18	22,7	1004	65,6	20100	600	10
127	FICH128.CSV	22/06/2020	9:31:53	22,9	1003	64,8	20098	600	12
128	FICH129.CSV	22/06/2020	10:30:30	22,9	1005	63	20104	600	14
129	FICH130.CSV	22/06/2020	11:29:06	23,4	1004	63,8	20095	600	11
130	FICH131.CSV	22/06/2020	12:27:43	23,3	1004	65,6	20101	600	11
131	FICH132.CSV	22/06/2020	13:26:20	23,6	1004	65,1	20083	600	13



ANEXO V

HOJAS DE DATOS

5,000ppm Carbon Monoxide Electrochemical Sensor

This carbon monoxide sensor is designed to provide students or scientists with proven electrochemical technology. OEMs will appreciate the low power, dependable technology for use in high volume applications.

FEATURES

- Low power
- Low cost
- Serial interface
- Replaceable electrochemical cell

AVAILABLE MODELS

CM-31911 Development Kit – Our easiest to use version, it is ready to plug into your PC via USB. Use our free GasLab[®] software to read and data log carbon monoxide, temperature and % relative humidity.

CM-31910 – This PCB offers the same functionality as the development kit, but is designed to be integrated into your product.

AP-0005 – Raw sensor only



ELECTRICAL SPECIFICATIONS	
Supply Voltage	3.3 to 5.5 VDC
Peak Supply Current	10ma
Average Power	< 3mW (1 second streaming 1 min logging)
UART Tx	3V 9600 Baud N 8 1
UART Rx Voltage	3V – 5V
Operating Temp	-30°C – 60°C
Humidity	0 – 99% RH (non-condensing)
Barometric Pressure	50 – 115 kpa

CONNECTOR PINOUT*	
GND	SELECT
3.3 – 5.5 Volt DC	GND
Rx	GND
Tx	RS485 B
Analog Out	RS485 A

MECHANICAL SPECIFICATIONS	
Dimensions	25mm x 40mm x 29mm
Dimensions (w/sensor)	25mm x 40mm x 13mm
Weight	6g sensor, 5.8g board
Connector	10 pin Header

ABSOLUTE MAXIMUM RATINGS (EC3 Controller Only)	
Supply Voltage	6 Volts DC
Rx Input	5.5 V
Operating Temp	-30°C – 60°C
Humidity	0 – 99% RH (non-condensing)
Pressure	500 kpa

COMMON PERFORMANCE SPECIFICATIONS	
Temperature Accuracy at 20-40°C	±0.3°C
Temperature Accuracy 0 – 50°C	±2% C
Relative Humidity Accuracy 20% – 80%	±2% RH
Sensor Voltage Resolution	16 bits

SIGNAL DEFINITIONS	
GND	Power Supply and RS485 return.
+SUPPLY	+3.2 – 5.5 Volts DC
Rx	CMOS Level Input to Controller
Tx	CMOS Level (0-3V) Output from controller.
AOUT	Analog Output from controller (when enabled). Voltage is proportional to gas concentration.
SELECT	Open selects the CMOS Tx/Rx Interface. Connect to GND to select the RS485 Interface.
RS485 B	RS485 B Signal. High in Marking State
RS485 A	RS485 A Signal. Low in Marking State.

AlphaSense O2-A2 Oxygen Sensor

This oxygen sensor is designed to provide students or scientists with proven electrochemical technology. OEMs will appreciate the low power, dependable technology for use in high volume applications.

FEATURES

- Low power
- Low cost
- Serial interface
- Replaceable electrochemical cell

AVAILABLE MODELS

CM-32911 Development Kit – Our easiest to use version, it is ready to plug into your PC via USB. Use our free GasLab[®] software to read and data log oxygen, barometric pressure, temperature, and % relative humidity.

CM-32910 – This PCB offers the same functionality as the development kit, but is designed to be integrated into your product.

AP-0001 – Raw sensor only



ELECTRICAL SPECIFICATIONS	
Supply Voltage	3.3 to 5.5 VDC
Peak Supply Current	10ma
Average Power	< 3mW (1 second streaming 1 min logging)
UART Tx	3V 9600 Baud N 8 1
UART Rx Voltage	3V – 5V
Operating Temp	-30C – 60C
Humidity	0 – 99% RH (non-condensing)
Barometric Pressure	50 – 115 kpa

CONNECTOR PINOUT*	
GND	SELECT
3.3 – 5.5 Volt DC	GND
Rx	GND
Tx	RS485 B
Analog Out	RS485 A

MECHANICAL SPECIFICATIONS	
Dimensions	25mm x 40mm x 29mm
Dimensions (w/sensor)	25mm x 40mm x 13mm
Weight	16g sensor, 5.8g board
Connector	10 pin Header

ABSOLUTE MAXIMUM RATINGS (EC3 Controller Only)	
Supply Voltage	6 Volts DC
Rx Input	5.5 V
Operating Temp	-30°C – 60°C
Humidity	0 – 99% RH (non-condensing)
Pressure	500 kpa

COMMON PERFORMANCE SPECIFICATIONS	
Temperature Accuracy at 20-40°C	±0.3°C
Temperature Accuracy 0 – 50°C	±2%
Relative Humidity Accuracy 20% – 80%	±2% RH
Sensor Voltage Resolution	16 bits

SIGNAL DEFINITIONS	
GND	Power Supply and RS485 return.
+SUPPLY	+3.2 – 5.5 Volts DC
Rx	CMOS Level Input to Controller
Tx	CMOS Level (0-3V) Output from controller.
AOUT	Analog Output from controller (when enabled). Voltage is proportional to gas concentration.
SELECT	Open selects the CMOS Tx/Rx Interface. Connect to GND to select the RS485 Interface.
RS485 B	RS485 B Signal. High in Marking State
RS485 A	RS485 A Signal. Low in Marking State.

OXYGEN SENSOR PERFORMANCE DATA

Figure 2 Output Temperature Dependence

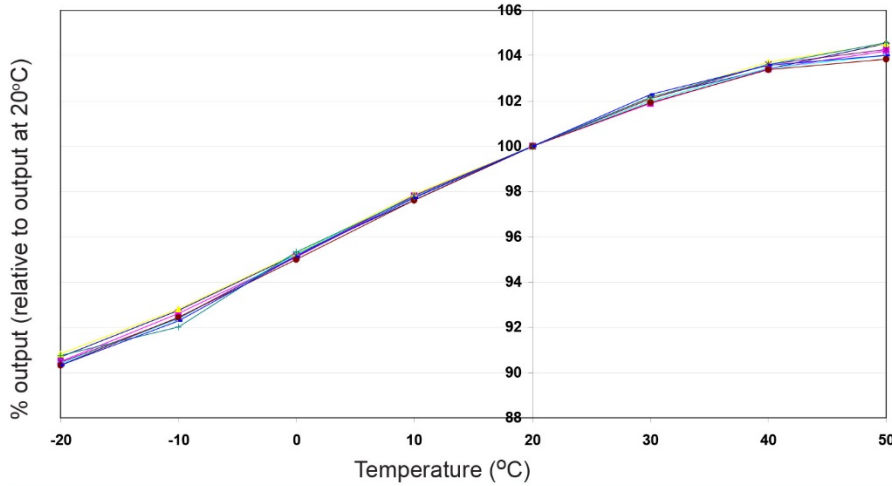
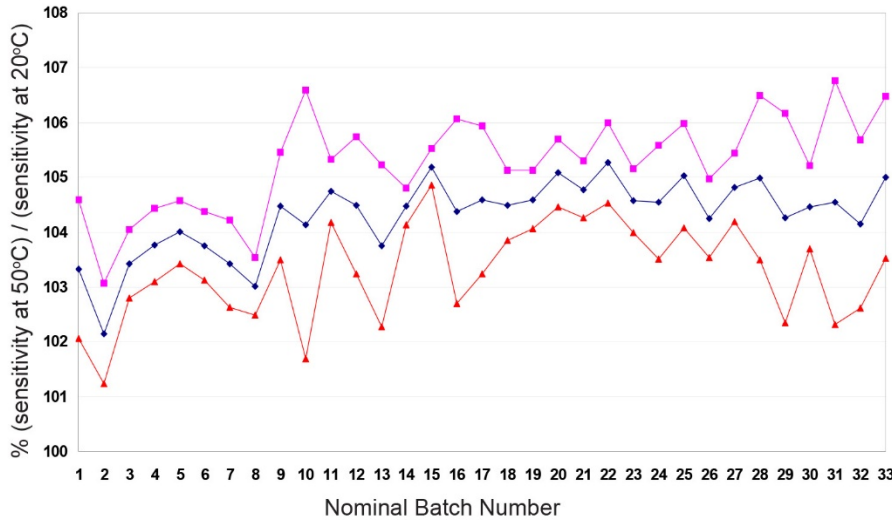


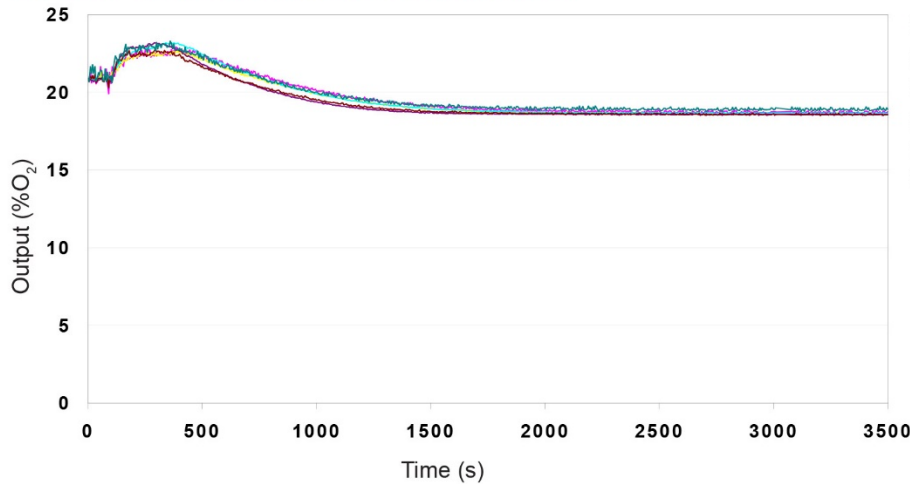
Figure 2 shows the variation in sensitivity caused by changes in temperature. Temperature dependence is very repeatable.

Figure 3 Sensitivity at 50°C



This plot of the mean and $\pm 95\%$ confidence intervals for 34 batches shows superior repeatability of the sensitivity dependence from batch to batch, giving confidence when setting temperature compensation in your gas detector.

Figure 4 Thermal Transient Performance



Sensors were thermally shocked from 20°C to -30°C. Consistent manufacture and good design ensure that there are no thermal spikes which can cause an alarm.

ExplorIR[®]-M

- **Miniature footprint CO₂ Sensor**
- **Ideal for battery-powered applications**
- **Fit and forget, fully autonomous operation**
- **Long life, >15 years**



About the ExplorIR[®]-M

ExplorIR[®]-M is a miniature CO₂ sensor, capable of measuring up to 100% concentration. Its compact and robust design enables easy integration into gas monitoring and detection systems.

ExplorIR[®]-M is specifically designed for applications that require the sensor to operate reliably in extreme environmental conditions, especially where the pressure, temperature or vibration regime is particularly harsh.

The ExplorIR[®]-M uses GSS patented solid-state optical technology, which enables the sensor to provide high accuracy CO₂ measurement capability over an extended lifetime.

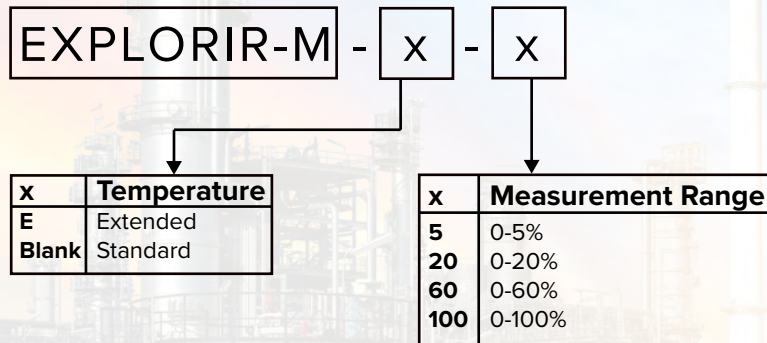
Features

- Measure up to 100% CO₂ concentration
- Low power CO₂ sensor
- Solid state LED optical technology
- Vibration and shock resistant
- UART data interface
- Built-in auto-calibration

Applications

- Industrial Safety
- Incubators
- Transportation
- Refrigeration
- Horticulture and Agriculture

Ordering Information



CO₂ Sensor Specifications

Measurement Ranges	0-5%, 0-20%, 0-60%, 0-100%
Accuracy (typ.)	0-60% ±(70ppm +5% of reading) 0-100% ±(300ppm +5% of reading)
Time to 1st Reading	1.2 seconds
Response Time	<30 Seconds (Diffusion limited)
Readings per Second	2
Sample Method	Solid-state LED NDIR Diffusion

Electrical and Mechanical Specifications

Measurement Output	UART
Supply Voltage	3.25V to 5.5V
Power Consumption (typ.)	3.5mW @ 3.3V
Dimensions and Weight	ø20.9mm x 18.1mm, 4g

Operating Conditions

Operating Conditions - Temperature	0°C to 50°C (Standard) -25°C to 55°C (Extended)
Operating Conditions - Humidity	0-95% RH, non-condensing
Storage Conditions - Temperature	-30°C to +70°C
Pressure Dependence	500mbar - 10bar
Sensor Lifetime	>15 years
Environmental Compliance	RoHS and REACH

Certificate of Registration

QUALITY MANAGEMENT SYSTEM - ISO 9001:2015

This is to certify that:

Gas Sensing Solutions
Grayhill Road
Cumbernauld
Glasgow
G68 9HQ
United Kingdom

Holds Certificate Number:

FM 585046

and operates a Quality Management System which complies with the requirements of ISO 9001:2015 for the following scope:

Design and manufacture of gas sensors for industrial and commercial applications.

For and on behalf of BSI:



Andrew Launn, EMEA Systems Certification Director

Original Registration Date: 2012-05-15

Latest Revision Date: 2018-05-09

Effective Date: 2018-05-16

Expiry Date: 2021-05-15



003

Page: 1 of 2

...making excellence a habit.™

Certificate No: FM 585046

Location	Registered Activities
Gas Sensing Solutions Grayhill Road Cumbernauld Glasgow G68 9HQ United Kingdom	Design and manufacture of gas sensors for industrial and commercial applications.
Gas Sensing Solutions T/A Quantum Device Solutions Block 7 Kelvin Campus West Scotland Business Park Glasgow G20 0SP United Kingdom	Design, development and manufacture of Group III-V semiconductor devices.



Original Registration Date: 2012-05-15

Latest Revision Date: 2018-05-09

Effective Date: 2018-05-16

Expiry Date: 2021-05-15

Page: 2 of 2

This certificate was issued electronically and remains the property of BSI and is bound by the conditions of contract.
An electronic certificate can be authenticated [online](#).
Printed copies can be validated at www.bsigroup.com/ClientDirectory

Information and Contact: BSI, Kitemark Court, Davy Avenue, Knowlhill, Milton Keynes MK5 8PP. Tel: + 44 345 080 9000
BSI Assurance UK Limited, registered in England under number 7805321 at 389 Chiswick High Road, London W4 4AL, UK.
A Member of the BSI Group of Companies.

8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash

DATASHEET

Features

- High performance, low power AVR® 8-bit microcontroller
- Advanced RISC architecture
 - 131 powerful instructions – most single clock cycle execution
 - 32 × 8 general purpose working registers
 - Fully static operation
 - Up to 16MIPS throughput at 16MHz
 - On-chip 2-cycle multiplier
- High endurance non-volatile memory segments
 - 32K bytes of in-system self-programmable flash program memory
 - 1Kbytes EEPROM
 - 2Kbytes internal SRAM
 - Write/erase cycles: 10,000 flash/100,000 EEPROM
 - Optional boot code section with independent lock bits
 - In-system programming by on-chip boot program
 - True read-while-write operation
 - Programming lock for software security
- Peripheral features
 - Two 8-bit Timer/Counters with separate prescaler and compare mode
 - One 16-bit Timer/Counter with separate prescaler, compare mode, and capture mode
 - Real time counter with separate oscillator
 - Six PWM channels
 - 8-channel 10-bit ADC in TQFP and QFN/MLF package
 - Temperature measurement
 - Programmable serial USART
 - Master/slave SPI serial interface
 - Byte-oriented 2-wire serial interface (Phillips I²C compatible)
 - Programmable watchdog timer with separate on-chip oscillator
 - On-chip analog comparator
 - Interrupt and wake-up on pin change
- Special microcontroller features
 - Power-on reset and programmable brown-out detection
 - Internal calibrated oscillator
 - External and internal interrupt sources
 - Six sleep modes: Idle, ADC noise reduction, power-save, power-down, standby, and extended standby

Para más información acerca de este componente, su Datasheet completo se encuentra en la URL:

http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf