



UNIVERSITAT  
JAUME·I

UNIVERSIDAD JAUME I  
ESCOLA SUPERIOR DE TECNOLOGIA I CIÈNCIES  
EXPERIMENTALS

GRADO DE INGENIERÍA ELÉCTRICA

*Diseño e implementación de una tarjeta  
electrónica para el control de una pila de  
combustible de hidrógeno*

**TRABAJO DE FIN DE GRADO**

AUTOR/A

Adrián Carrión Tellols

DIRECTOR/A

Ignacio Peñarrocha Alós

Castellón, Agosto 2020



# ÍNDICE

1. Memoria.....	5
1.1 Objeto.....	7
1.2 Alcance.....	7
1.3 Antecedentes.....	8
1.4 Características pila de combustible de hidrógeno.....	10
1.5 Normas y referencias.....	10
1.5.1 Disposiciones legales y normas aplicadas.....	10
1.5.2 Programas de cálculo.....	11
1.5.3 Bibliografía.....	13
1.5.4 Otras referencias.....	14
1.6 Definiciones y abreviaturas.....	14
1.7 Requisitos de diseño.....	15
1.8 Análisis de soluciones.....	16
1.9 Resultados finales.....	26
1.10 Planificación.....	30
1.11 Orden de prioridad entre los documentos.....	31
2. Anexos.....	33
2.1 Contenido.....	35
2.1.1 Cálculos.....	35
2.1.2 Estudio básico de seguridad y salud.....	39
2.1.3 Hojas de datos.....	41
2.1.4 Código arduino.....	62
2.1.5 Código codesys.....	69
2.1.6 Shield Arduino.....	78
3. Planos.....	84
4. Pliego de condiciones.....	89
4.1 Requisitos necesarios para la realización del proyecto.....	91
4.2 Manual de instrucciones.....	91
5. Mediciones.....	93
6. Presupuesto.....	98



# **MEMORIA**



## 1.1 OBJETO

El siguiente proyecto tiene como finalidad el diseño e implementación de un Shield para una tarjeta Arduino Due, con el cual se controlará un banco de pruebas de pilas de combustible de hidrógeno tipo PEM.

Este proyecto pretende cubrir las necesidades propuestas por la Universidad CEU Cardenal Herrera, en la cual el proyectista realizó las prácticas.

Debido a la complejidad del banco de pruebas, el proyectista y el promotor acordaron que dicho TFG resolviera las demandas de control de potencia y medida de datos para ello.

## 1.2 ALCANCE

El ámbito de este proyecto es la automatización, ya que todo el proyecto se va a basar en la programación de una tarjeta diseñada para que el usuario pueda controlar de manera sencilla las condiciones en las que se estudia la pila de combustible de hidrógeno.

Este proyecto es muy específico y al pretender solucionar las demandas propuestas por la CEU, está pensado para que su ámbito de aplicación sea en un laboratorio de investigación sobre pilas de combustible de hidrógeno.

Otra aplicación podría ser docente, ya que el programa se podría utilizar en una simulación y puede enseñar conocimientos de comunicación Maestro-Esclavo entre PC y tarjeta Arduino a parte de refrescar conceptos de programación vistos en ingeniería.

Por otra parte, los siguientes puntos no alcanza dicho proyecto:

- Estructura mecánica para la realización de las pruebas descritas en el proyecto.
- Alimentación regulable para cubrir la necesidad de algunos de los componentes descritos en el proyecto.
- Proyecto que incluya la lectura de temperaturas, voltajes e intensidades de la pila y su estudio.
- No estudiará los valores que se deben de utilizar en los controladores, pero quedará señalado para su uso.

### 1.3 ANTECEDENTES

El proyecto que ocupa, nace de la necesidad de poder experimentar con pilas de combustible de hidrógeno en el laboratorio tecnológico de la Universidad CEU Cardenal Herrera.

Una pila de combustible es un dispositivo electroquímico que transforma de forma directa la energía química en eléctrica. Para ello utiliza dos componentes, hidrógeno y oxígeno puros y tiene como salida agua. En concreto, las de tipo PEM consisten en una membrana de intercambio de protones donde los combustibles son hidrógeno y oxígeno por otro lado, dando como resultado electricidad y de residuo agua. Actualmente esta tecnología alcanza una conversión de energía química-eléctrica con un rendimiento del 60%, con posibilidad de mejora según los expertos.

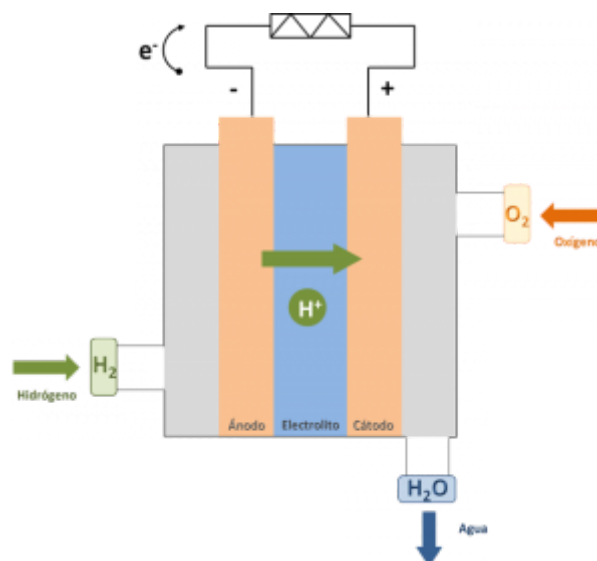


Ilustración 1: Dibujo simplificado pila PEM

En la actualidad que vivimos, la contaminación en nuestro planeta hace replantearnos el uso de combustibles fósiles, los cuales generando  $CO_2$  entre otros gases de efecto invernadero están consiguiendo subir la temperatura global del planeta. Para evitar este fenómeno se estudian diferentes alternativas energéticas, como coches eléctricos con baterías de alta carga, automóviles híbridos... Y una de esas alternativas son las pilas de hidrógeno, que ya se utilizan en automóviles.



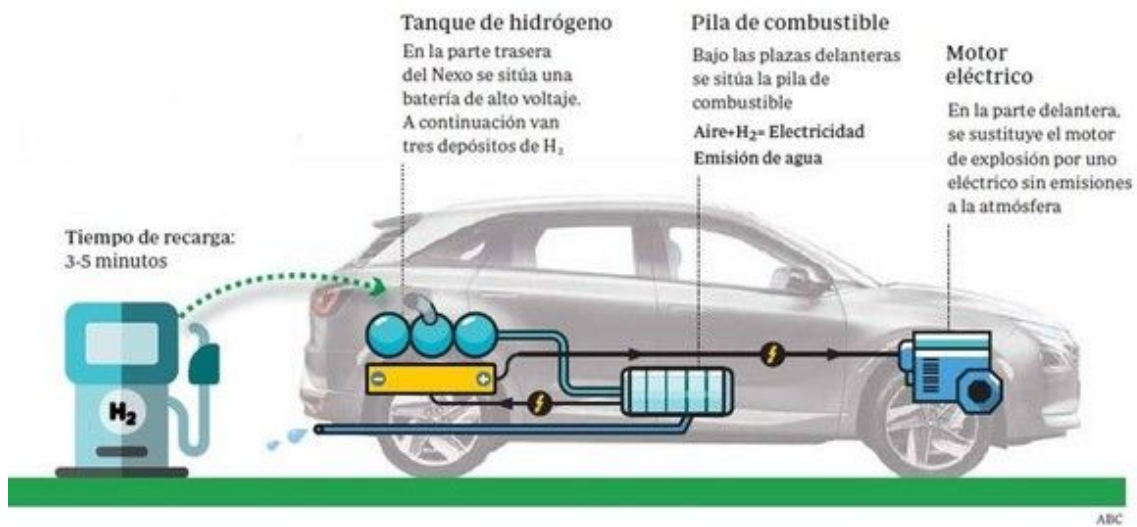


Ilustración 2: Imagen coche de combustible hidrógeno

Actualmente su mayor rival en el sector es el coche eléctrico, ya que ambos ofrecen una solución ecológica al problema del coche con motor de explosión. Ambas tecnologías funcionan con un motor eléctrico, solo que en uno se obtiene la energía eléctrica de una batería y en el segundo por la reacción química entre el oxígeno y el hidrógeno mientras forman el agua.

Aquí una breve comparativa de ambas tecnologías, donde se muestra la eficiencia del coche eléctrico, del coche con pila de hidrógeno y el clásico coche con motor de explosión.

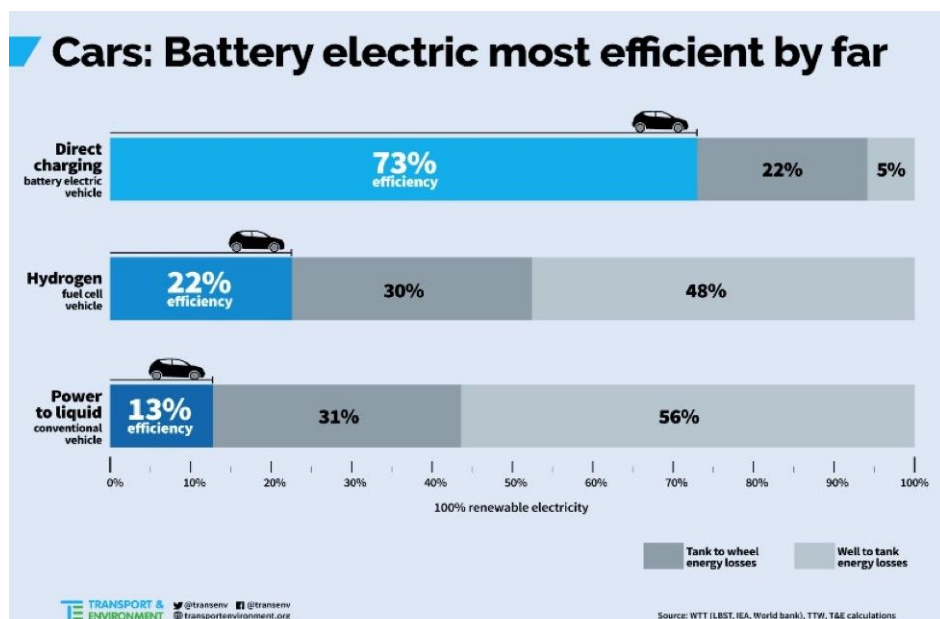


Ilustración 3: Imagen comparativa coche con batería, coche con hidrógeno y coche de gasolina

De la comparativa anterior y del calentamiento global nace la necesidad de estudiar las pilas de combustible de hidrógeno. Actualmente su tecnología es inferior a la eléctrica, pero sigue siendo renovable en un mundo donde la contaminación es un problema grave. De dicha necesidad nace este proyecto, el cual controlará las condiciones en las que la pila de hidrógeno es estudiada en el laboratorio de la CEU para su posible mejora.

#### 1.4 CARACTERÍSTICAS PILA DE COMBUSTIBLE DE HIDRÓGENO

El proyectista y el promotor acordaron en el diseño que la pila de combustible debía de regirse por el siguiente esquema, consta de 2 válvulas de apertura y cierre más 2 de purga, iguales entre ellas. 2 Reguladores máscicos que controlan el caudal y presión de entrada y dos pares de sensores de presión y caudal para medir el sistema. El caudal de entrada del regulador no superará los 50 l/min tanto en Hidrógeno como en Oxígeno. De la misma manera no superará los 250 psi o 17,2 bar de presión. Los sensores de presión y caudal se seleccionarán en un rango próximo a los valores de salida del regulador máscico.

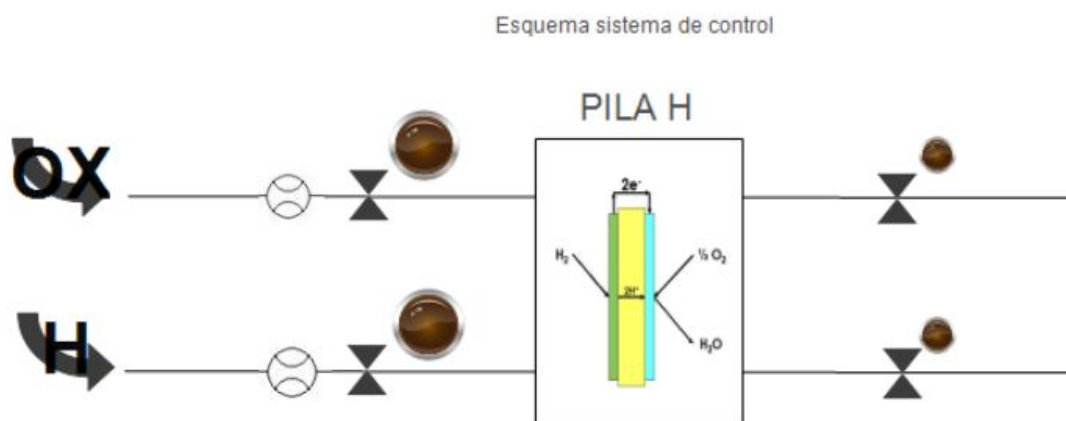


Ilustración 4: Esquema general pila de hidrógeno tipo PEM realizado por el proyectista

#### 1.5 NORMAS Y REFERENCIAS

##### 1.5.1 DISPOSICIONES LEGALES Y NORMAS APLICADAS \*

La redacción de este proyecto ha seguido las bases establecidas por las siguientes normativas:

- UNE1570010: Criterios generales para la elaboración formal de los documentos que constituye un proyecto técnico.
- UNE-EN ISO 9000: Sistemas de gestión de la calidad. Fundamentos y vocabulario.
- UNE 1032:1982: Dibujos técnicos. Principios de representación.
- ISO 45001: Seguridad.

## 1.5.2 PROGRAMAS DE CÁLCULO

Eagle 9.5.2: Es un software de diseño de placas de circuito impreso(PCBs) muy utilizado en el entorno de la industria. La política del programa es privativa, aunque dispone de una versión limitada gratuita la cual permite diseñar placas de hasta 80 centímetros cuadrados. Contiene muchas librerías y diseños de componentes hechos por la comunidad. También es posible diseñar uno mismo los componentes de la PCB.

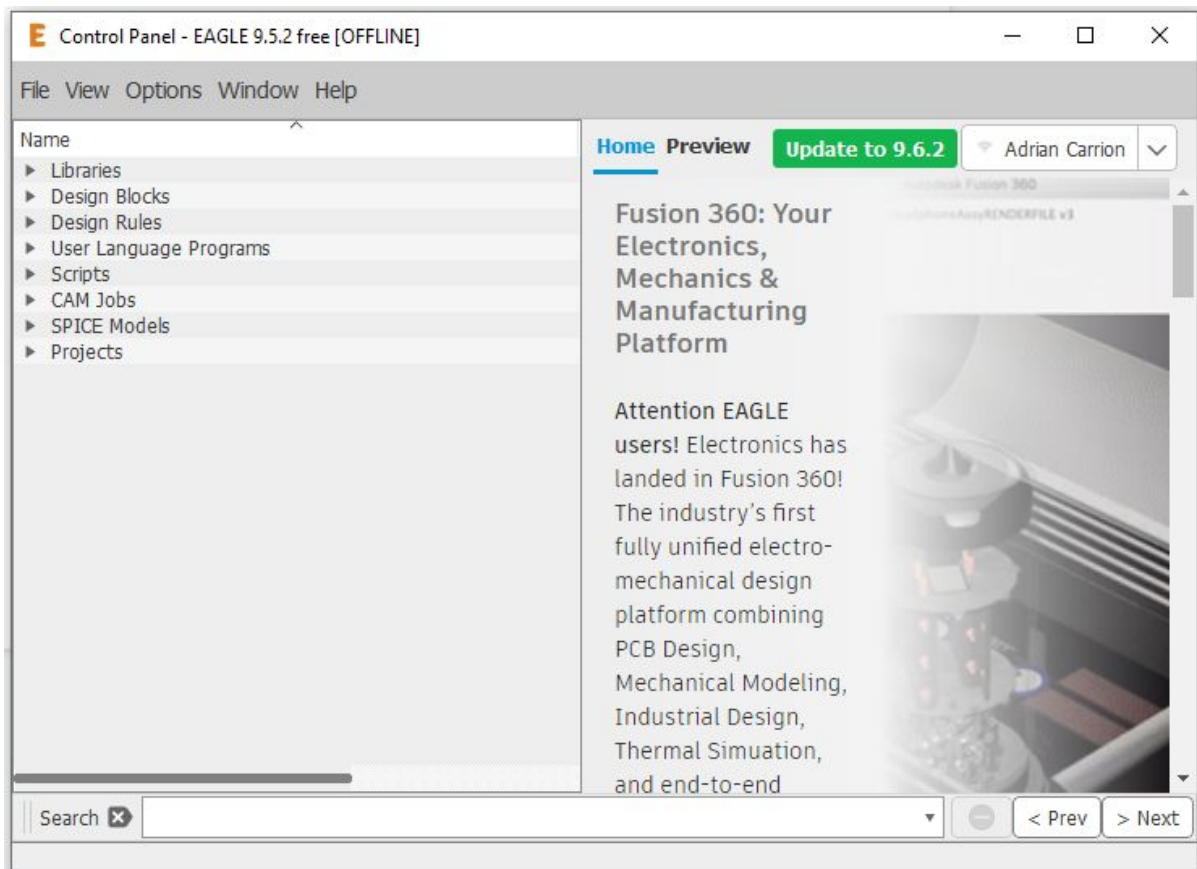


Ilustración 5: Captura de pantalla del programa Eagle

Arduino IDE 1.8.12: Es una aplicación multiplataforma donde el usuario escribe y compila el código del programa. Es de uso libre y conocido en el ámbito de la ingeniería, tanto en la docencia, investigación e industria. Admite lenguajes tipo C y C ++, con una serie de reglas especiales de estructuración. Al ser libre, existen infinidad de librerías las cuales son muy útiles en cualquier proyecto.

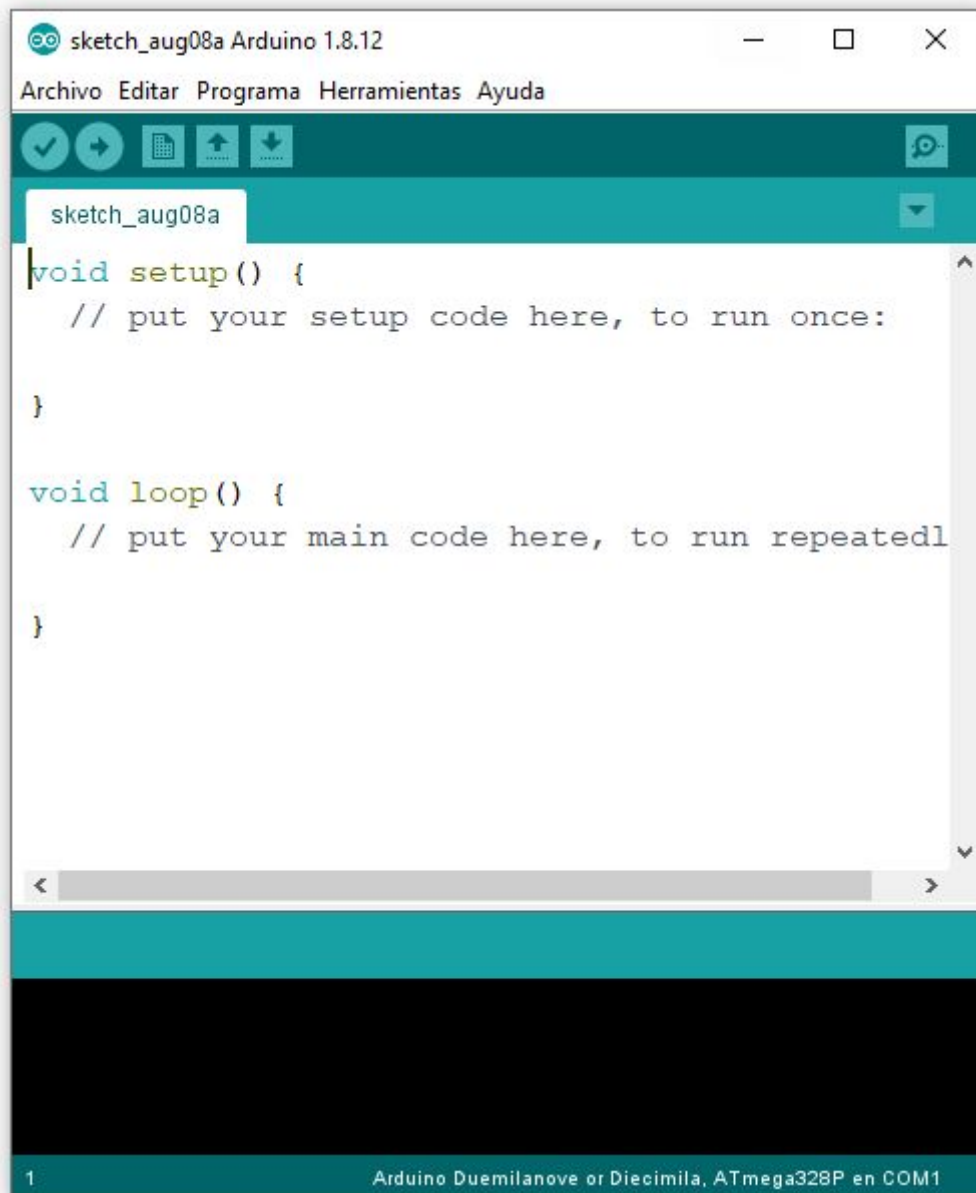


Ilustración 6: Captura de pantalla del programa Arduino

CODESYS V3.5 SP15 Patch 4: Se trata de un software industrial para la programación de controladores siguiendo el estándar IEC 61131-3. Codesys funciona con 6 lenguajes de programación. Las diversas funciones del programa hacen que sea la opción más utilizada en Europa actualmente en el mundo de la industria. No es un software libre.

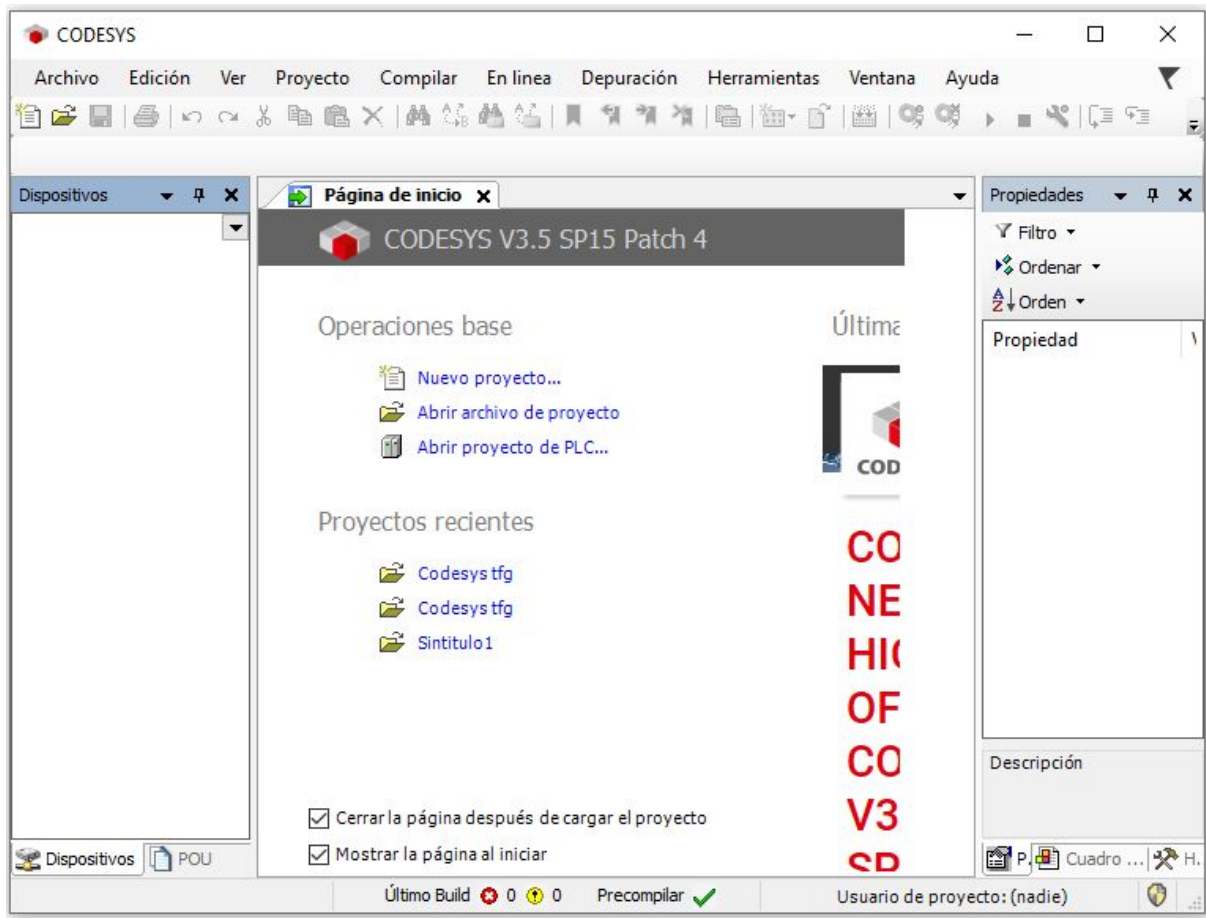


Ilustración 7: Captura de pantalla del programa Codesys

### 1.5.3 BIBLIOGRAFÍA

Apuntes asignatura EE 1003 Informática.

Apuntes asignatura EE 1018. Electrónica.

Apuntes asignatura EE 1030. Automatización industrial.

Apuntes asignatura EE 1031. Proyectos de ingeniería.

Apuntes asignatura EE 1035. Instrumentación medida y tratamiento de la señal.

Apuntes asignatura EE 1036. Programación de sistemas.

Apuntes asignatura EE 1038. Regulación automática.

Apuntes asignatura EE 1039. Sistemas informáticos industriales.

Web <https://www.arduino.cc/> Programación arduino

PDF [hetpro\\_tutorial\\_pcb\\_eagle.pdf](#). Diseño PDBs

#### 1.5.4 OTRAS REFERENCIAS

En este apartado se señalan otras referencias de interés que han aportado conocimientos para el desarrollo de este proyecto. Destaco estos videotutoriales:

Tutoriales de diseño EAGLE:

[https://www.youtube.com/watch?v=8Zdl-Ixke0w&t=1s&ab\\_channel=ElectroTutoriales](https://www.youtube.com/watch?v=8Zdl-Ixke0w&t=1s&ab_channel=ElectroTutoriales)

[https://www.youtube.com/watch?v=z\\_UKkBNz-iE&t=1s&ab\\_channel=ElectroTutoriales](https://www.youtube.com/watch?v=z_UKkBNz-iE&t=1s&ab_channel=ElectroTutoriales)

Tutoriales Codesys:

[https://www.youtube.com/watch?v=9zTYckmRp-I&t=686s&ab\\_channel=LeonardoRodriguezOrtiz](https://www.youtube.com/watch?v=9zTYckmRp-I&t=686s&ab_channel=LeonardoRodriguezOrtiz)

[https://www.youtube.com/watch?v=nWWo\\_P-7pJo&ab\\_channel=jbpilon02](https://www.youtube.com/watch?v=nWWo_P-7pJo&ab_channel=jbpilon02)

#### 1.6 DEFINICIONES Y ABREVIATURAS

Las abreviaturas utilizadas en este proyecto y su significado siguen en el siguiente listado:

**PDB:** Placa de circuito impreso

**CEU:** Universidad Cardenal Herrera ubicada en Valencia

**PID\*:** (controlador proporcional, integral y derivativo) Es un mecanismo de control simultáneo por realimentación ampliamente usado en sistemas de control industrial.

**PWM:** Modulación por ancho de impulsos. Gracias a la frecuencia de la señal es posible modificar la energía o información enviada.

**PIN:** Conector de la placa.

**mA:** miliamperios.

**V:** Voltios

**$\Omega$ :** Ohmios, medida básica de la resistencia eléctrica de un material.

**Shield:** placa diseñada para encajar en otra y suplir/añadir funciones.

## 1.7 REQUISITOS DE DISEÑO

El sistema a desarrollar debe de cumplir los siguientes requerimientos, impuestos por el promotor:

- Este proyecto consiste solamente en la parte de control y todo lo necesario para dicho control.
- El sistema consistirá en dos reguladores másicos que controlarán mediante un PID que llevan integrado el caudal y la presión con la que se suministra Oxígeno e Hidrógeno a la pila. Estos reguladores funcionarán con una intensidad de 4 a 20 mA.
- Existirán dos válvulas de apertura y cierre para controlar la marcha y paro del sistema. Dichas válvulas funcionan con 24V DC.
- También existirán dos válvulas para realizar purgas en el sistema, tanto automática como manualmente y funcionaran como las anteriormente descritas.
- Los sensores tanto de presión como de caudal proporcionan una señal hacia la placa que podrá ser de 4-20 mA o bien de 0-5V.
- Interfaz de control sencilla de utilizar para el usuario que englobe todas las condiciones de control descritas anteriormente y posteriormente y que haga un seguimiento a tiempo real del sistema.
- Elección de sensores y actuadores que se ajusten a las demandas de una pila de combustible de hidrógeno.

Restricciones interpuestas por el proyectista:

- Para mejorar la seguridad, el sistema tendrá incorporado un pulsador de emergencia en la sala del estudio y una palanca de emergencia en el panel de control, ambas activando el estado de emergencia que consiste en cerrar las válvulas en orden de mayor presión a menor presión.
- Diseñar la PCB en una sola, no varios SHIELDS.
- Que fuera posible la ampliación de la placa y/o la adición de nuevas funciones en ésta, tanto en hardware como en software.

## 1.8 ANÁLISIS DE SOLUCIONES

Las herramientas para solventar las demandas anteriores serán justificadas en este apartado del proyecto. En primer lugar se comentarán los software utilizados.

### *Diseño de la PCB:*

- **KiCad:** Programa de software libre para el diseño de tarjetas de circuito impreso. Está formado por un esquemático y luego por un entorno de diseño de placa, similar al sistema Eagle. También tiene ventaja de al ser un software libre, contiene muchas librerías creadas por usuarios y la capacidad de importar diseños de otros usuarios o programas similares. Por último incluye un visor 3D de la placa.



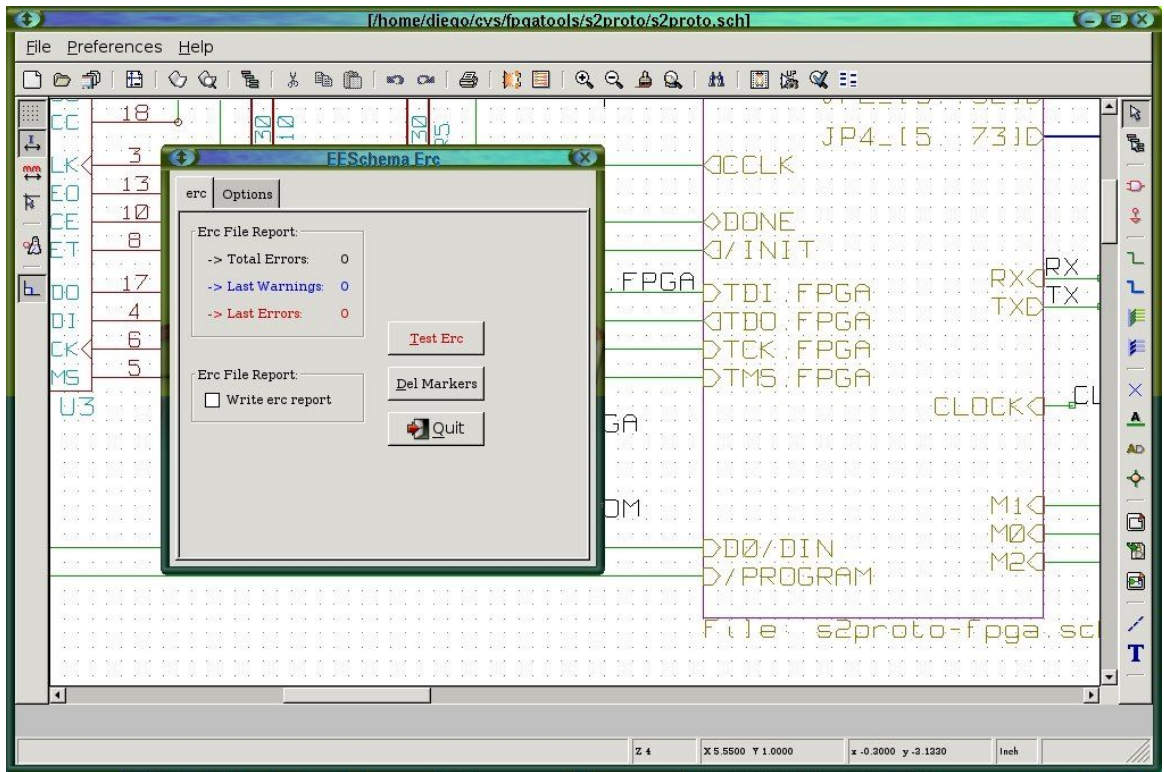


Ilustración 8: Captura de pantalla del programa KiCad, sección esquemático

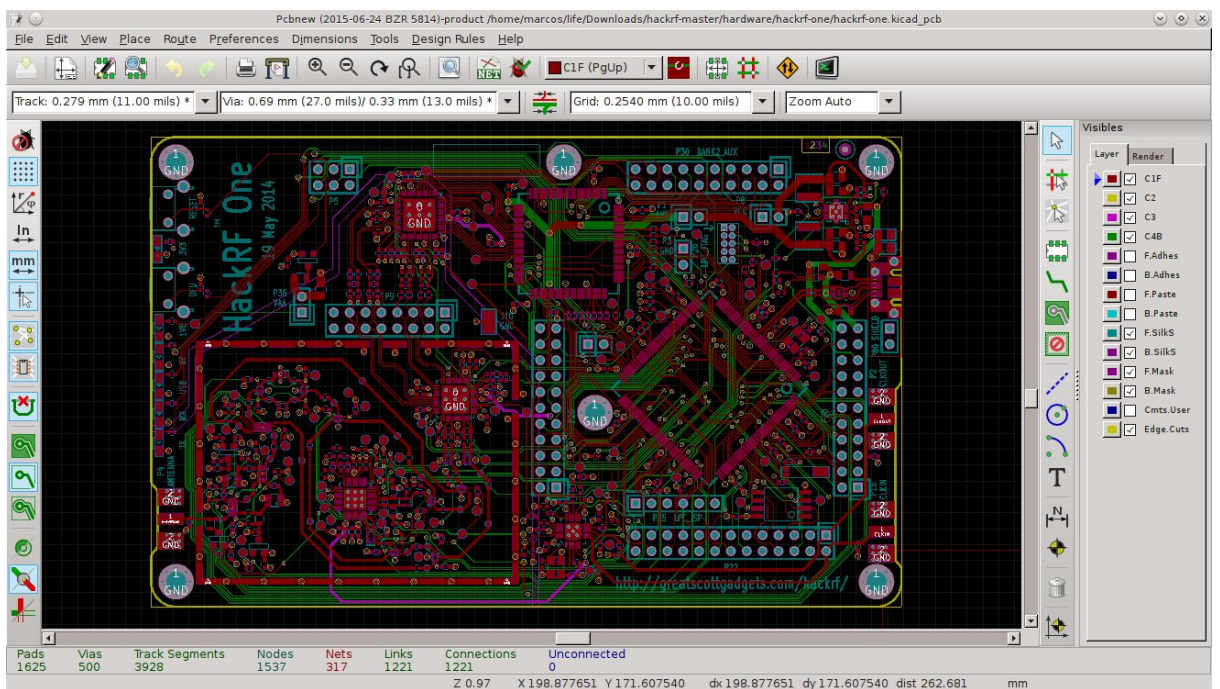


Ilustración 9: Captura de pantalla del programa KiCad, sección placa

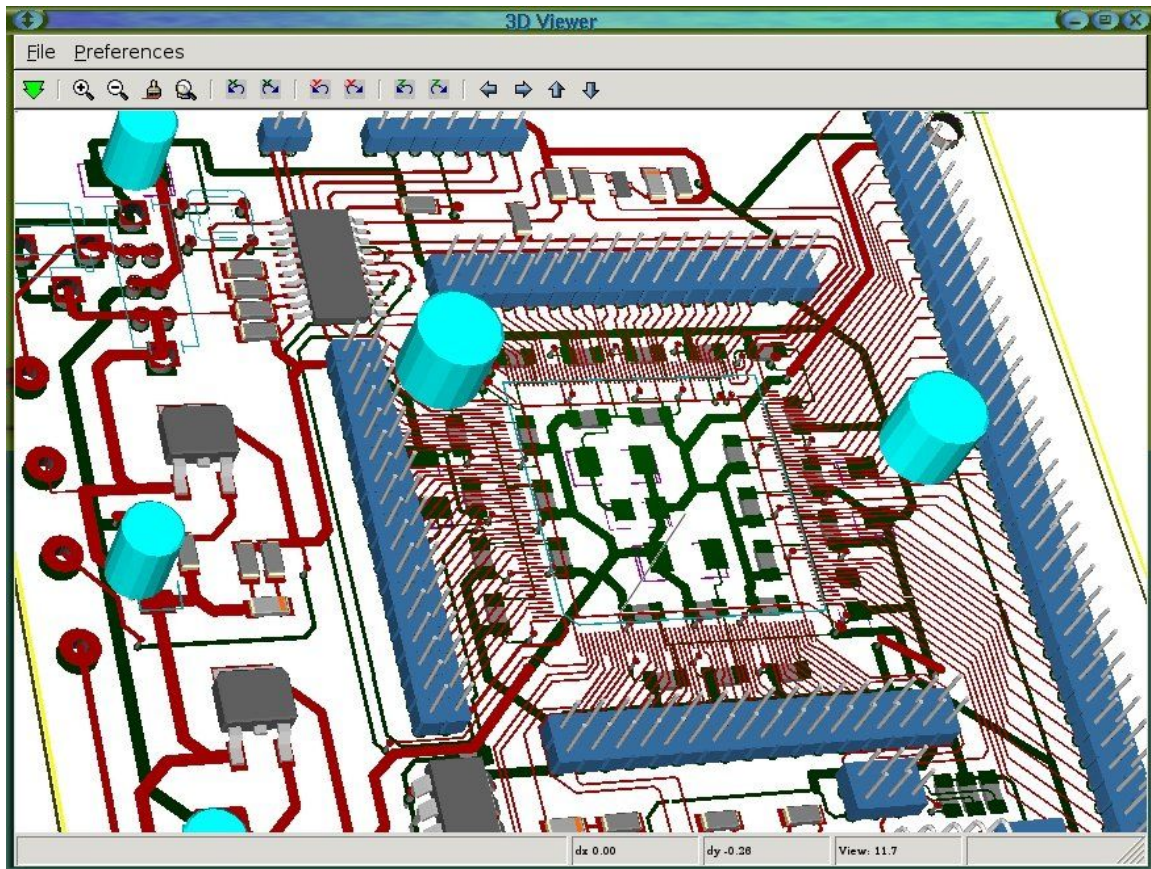


Ilustración 10: Captura de pantalla del programa KiCad, sección 3D

- **EasyEDA:** Es un conjunto de herramientas online, sin requisito de instalación para el diseño de PCB. Muy utilizado en la industria por la capacidad de sin tener instalado nada poder diseñar desde cualquier ordenador y dispone de la ventaja de que los archivos se pueden guardar en una nube. Es posible la importación de esquemas de otros programas, como Altium, Eagle, KiCad y LTspice. Su interfaz es muy simple y aunque tiene menos comandos es más sencillo que otros programas de utilizar.



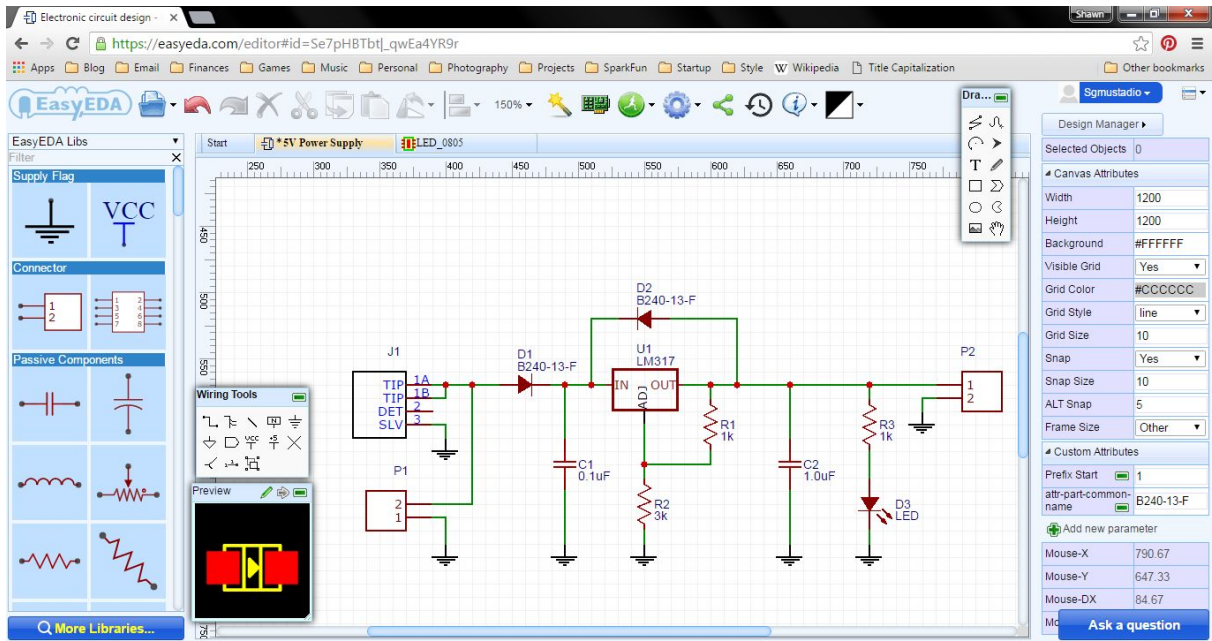


Ilustración 11: Captura de pantalla del programa EasyEDA, sección esquemático

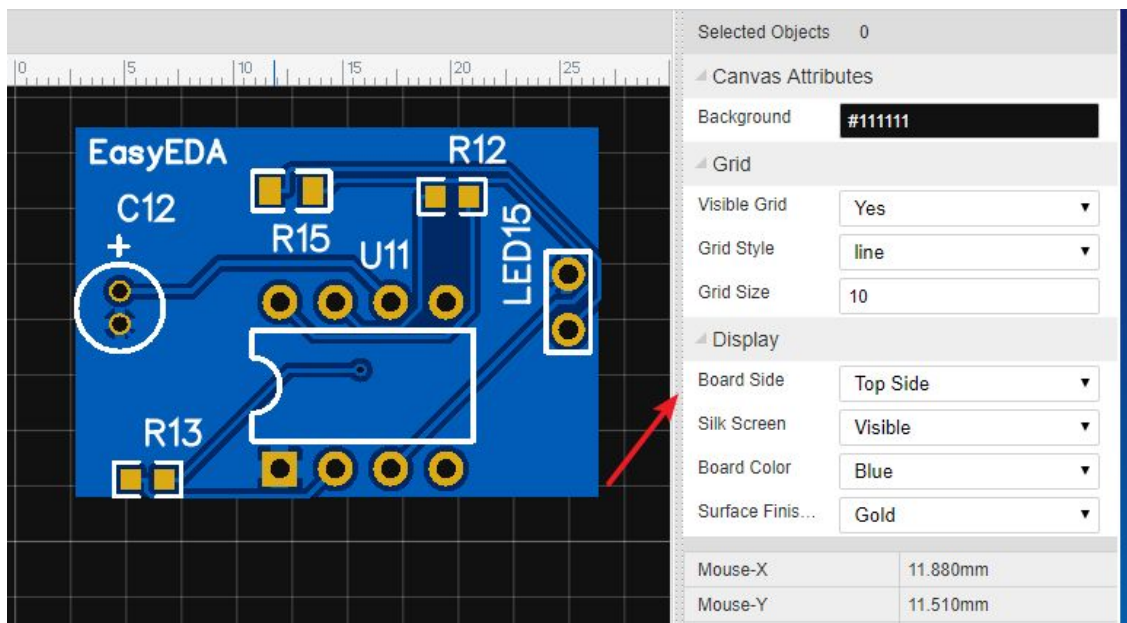


Ilustración 12: Captura de pantalla del programa EasyEDA, sección placa

- **Eagle:** Herramienta de diseño de circuitos impresos muy potente, utilizada bastante en proyectos de ingeniería, sobretodo en los más complejos. La gran cantidad de comandos y opciones lo convierten en un programa complejo de utilizar, y más cuando uno no está familiarizado con este tipo de programas pero un usuario dedicado puede sacar un alto partido al programa. Es posible importar diseños de otros

programas, librerías y modificaciones del programa, como por ejemplo añadir teclas de acceso rápido a las herramientas más utilizadas. Es el programa recomendado por el tutor también.

Como puntos malos del programa serían la alta complejidad para un usuario novato y que no es un programa gratuito, aunque dispone de una versión limitada libre.

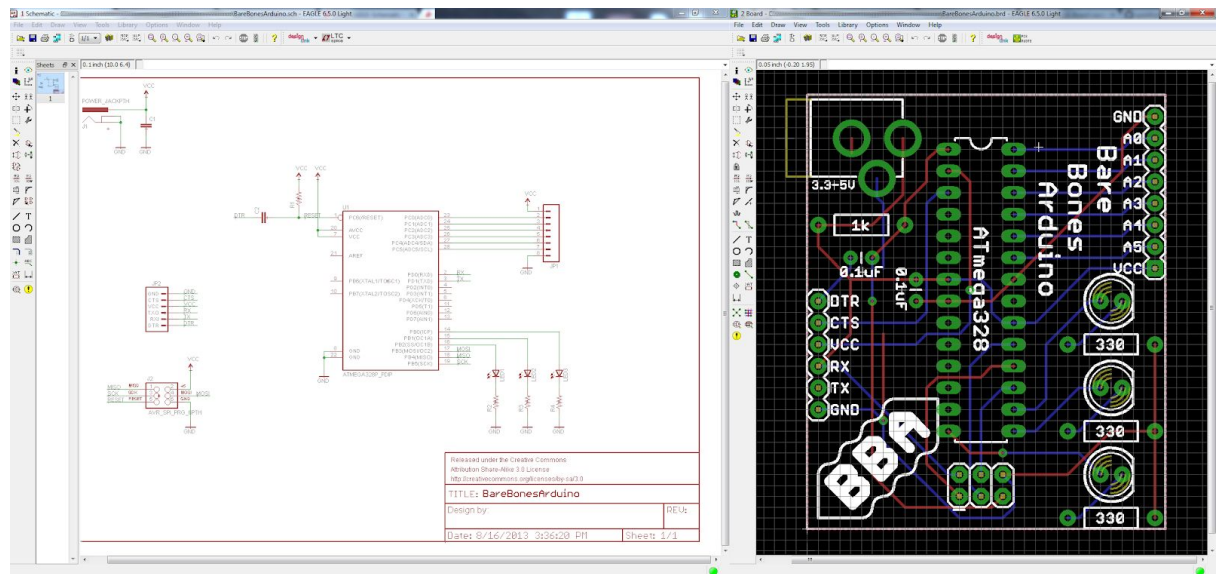


Ilustración 13: Captura de pantalla del programa EasyEDA, sección esquemático y placa

### *Programación de la placa arduino+shield:*

- **Arduino 1.8.12:** Sencillo, libre y muy utilizado en todo el ámbito ingenieril y sobretodo en el campo de la investigación. Al ser software libre, incluye muchas librerías creadas por usuarios que pueden ayudar al programador. Al haberlo estudiado mucho en mi formación como ingeniero, no existe razón para no utilizarlo.

### *Comunicación maestro esclavo PC-placa:*

- **Unity Pro 3.0:** Es un programa utilizado en la industria especialmente dedicado a autómatas Modicon, M340, Premium, Quantum y coprocesadores Atrium. Con una interfaz sencilla, dispone de herramientas para programar PLC. La principal ventaja son las estructuras que se incluyen en el programa que son muy utilizadas en estas aplicaciones y son usables/modificables por el programador, lo que ahorra tiempo. Dispone de secciones del programa lo que hace muy sencillo su uso y entendimiento.

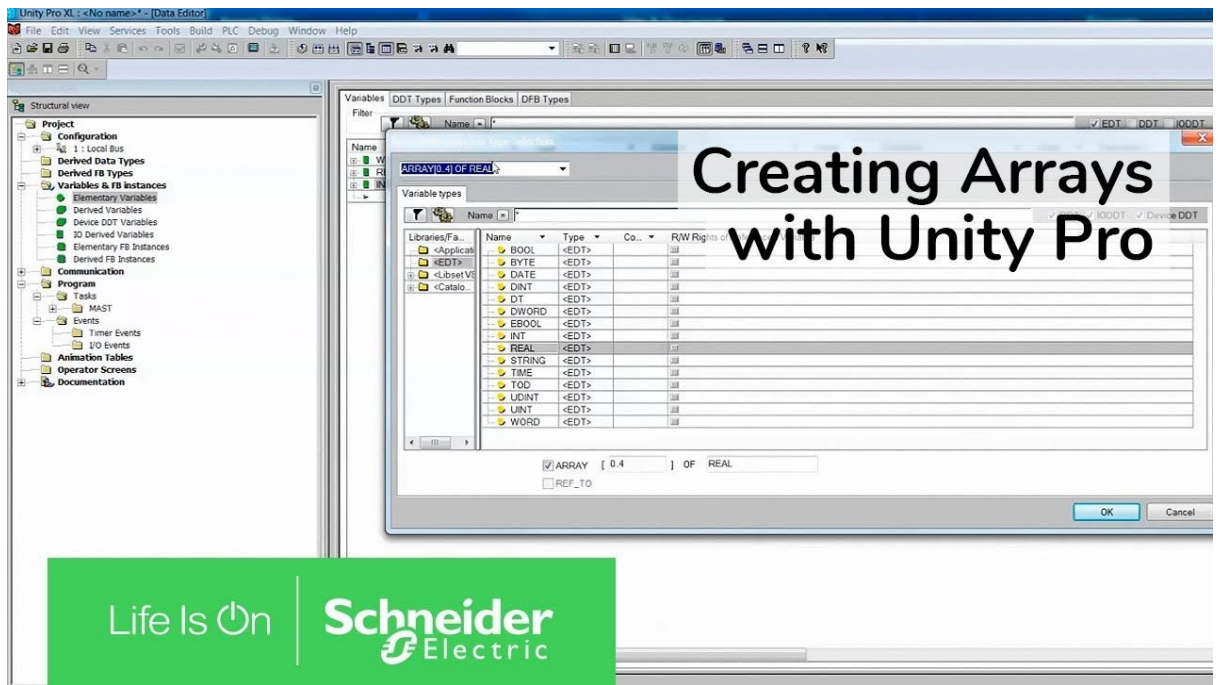


Ilustración 14: Captura de pantalla del programa Unity Pro

- **TriStation:** Es un software no libre en el cual se pueden programar PLC, aunque su mejor aplicación es en la seguridad de éstos. Se programa en diagrama de bloques de función, texto estructurado o diagrama de escalera. También dispone de una interfaz para arrastrar elementos. Con una gran cantidad de bloques para utilizar, este programa cumple con la IEC 61131-3.

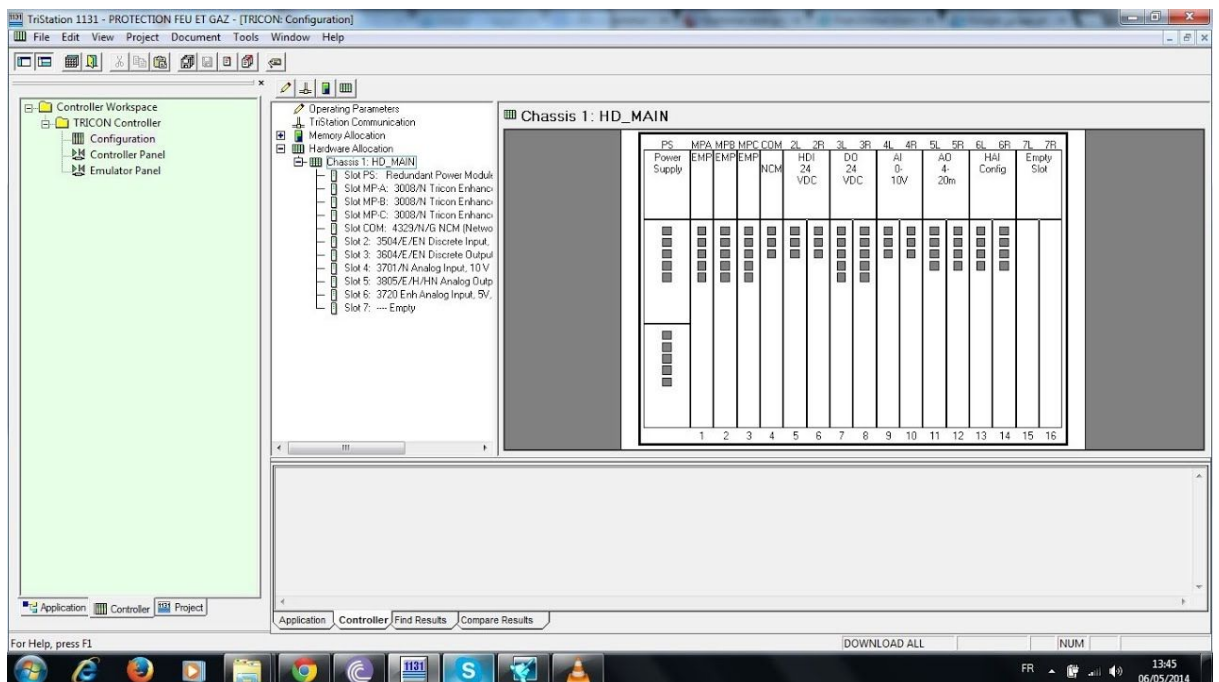


Ilustración 15: Captura de pantalla del programa TriStation

- **CODESYS V3.5 SP15 Patch 4:** Es el software no libre de programación de PLCs más utilizado en la industria en la zona europea. Dispone de varios lenguajes de programación, los cuales son LD, IL, FBD, SFC, ST, CFC. Combina estas potentes herramientas con otras de comunicación en tiempo real con el PLC o placa. A su vez el programa contiene una sección para crear interfaz de usuario, muy útil para que alguien que no ha diseñado el programa pueda utilizarlo con unas pocas pautas. Es la recomendación del tutor ya que en mi formación hemos aprendido a manejarlo.

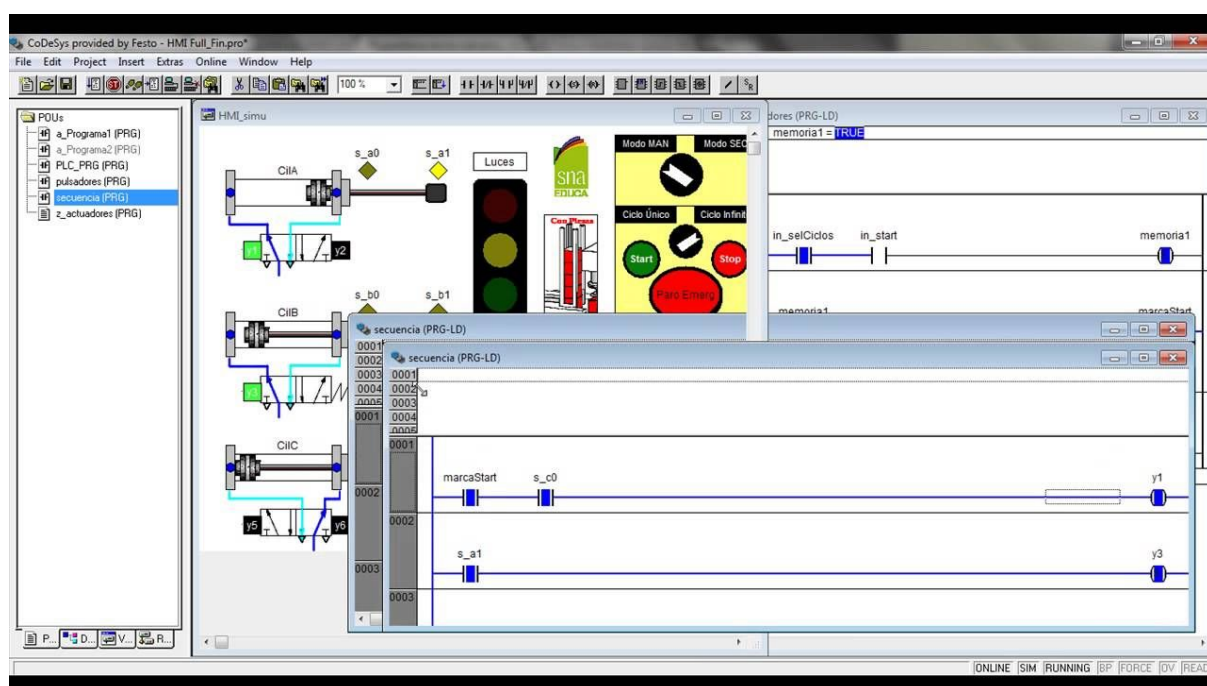


Ilustración 16: Captura de pantalla del programa Codesys

### *Diseño de un SHIELD:*

Para poder diseñar una PCB que cumpla con todo lo requerido se estudiará las limitaciones del programa de diseño y el espacio que ocupan los componentes necesarios. Con 80 centímetros cuadrados como máximo, los componentes también serán elegidos por su tamaño para cumplir con este requisito entre otros. También hay que tener en cuenta la disposición de pin que unen el shield y la placa arduino due para este diseño, por ejemplo unos contactores demasiado grandes no pueden ser puestos en la placa.

Existen diseños de módulos de contactores ya preparados para funcionar con placas de arduino como el Optoacoplador para Arduino UNO R3 MEGA 2560 Proyecto 1280 DSP ARM PIC AVR STM32 Raspberry Pi, pero el diseño de este imposibilita la adición de otros componentes para este proyecto como el control de los reguladores o la lectura de sensores. Aún así, el diseño sirve de ayuda para este proyecto:



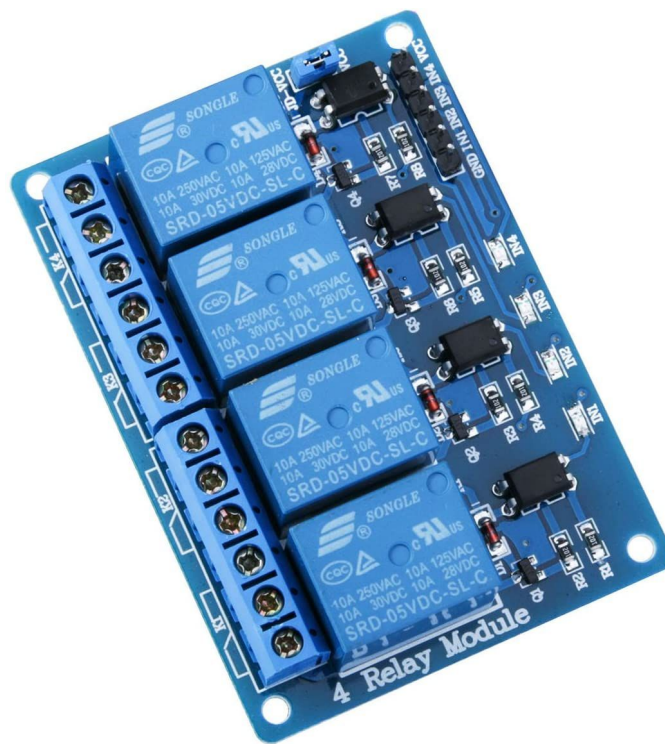


Ilustración 17: Optocoplador arduino de 4 contactores

Por otra parte los reguladores utilizarán la salida PWM que tiene el arduino due en sus pin, una señal cuadrada de frecuencia controlada por el arduino para convertirla en una señal continua con valores de 4 a 20mA. Esto será estudiado en el apartado de cálculos

Y por último los sensores pueden funcionar con un rango de 4-20mA o de 0 a 5V, también estudiado en el apartado de cálculos, donde se expondrán los circuitos y los cálculos hechos para llegar a ellos.

### *Ampliación:*

Aunque una placa impresa es difícilmente la adición de componentes, este proyecto como se verá en el apartado de cálculos, mediante la limitación de intensidad que puede dar la placa arduino y el número de pin que tiene, deja espacio para añadir funciones. Esto es muy útil ya que las pilas de combustible de hidrógeno son de muchos tipos y lo cual distintas necesidades, puede ser muy interesante la adición de algún componente para poder trabajar con otras.

El código arduino está estructurado de manera que hace sencilla la adición de más. A su vez en el codesys es sencillo añadir más funciones por lo modular que es el programa.

### *Regulador:*

Para el banco de pruebas, se seleccionan dos reguladores de caudal másico térmico SideTrak® 840, ya que las pilas de hidrógeno que se van a estudiar en dicho laboratorio necesitarán una presión máxima de 250 psi. El rango de presiones que ofrece este actuador es de 0 a 500 psi, muy amplio, perfecto para hacer muchos tipos de experimentos.

Por otro lado, el caudal cumple de manera muy holgada, teniendo un rango de operación de 0 a 500 l/min.

En los ANEXOS se adjunta la hoja de datos para su justificación.



Ilustración 18: Regulador Sidetrack 840

### *Sensores:*

Para el Caudal, se selecciona el caudalímetro SFAH, con varios rangos de operación, cuyo valor de medición máximo es de 200 l/min, un rango muy aceptable para el sistema en



cuestión. Cuenta con varios tipos de salida, entre otros la que más interesa, la de 4-20 mA pero el diseño de la placa de este proyecto admite 1-5V mediante unos Jumpers que incorpora.

En esta aplicación se usará el 8058474 SFAH-100U-Q8S-PNLK-PNVBA-L1, que va de 0 a 100 l/min.

En los ANEXOS se adjunta la hoja de datos para su justificación.



Ilustración 19: Sensor caudalimetro SFAH

Para la medición de la presión, se utilizarán dos sensores, uno para el oxígeno y otro para el hidrógeno. Serán los transductores de baja presión industrial SIL, con un rango de 0-40 bar, siendo el regulador que utiliza el proyecto de 0-35 bar cumple. También dispone de salida 4-20mA o de 1-5V, ambas salidas posibles en la PCB diseñada.



Ilustración 20: Sensor de baja presión industrial SIL

### *Válvulas:*

Son necesarias 4 válvulas para el control del sistema. Tanto las de purga como las de apertura y cierre son iguales aunque se controlen de manera diferente. Es necesario que funcionen con una tensión DC de 24V, que sean especialmente diseñadas para gas y que soporten una presión de 40 bar. Para ello se seleccionan las electroválvulas SIRAI L172V03 DC/CC, que soportan hasta 50 bar además de cumplir las demás exigencias.

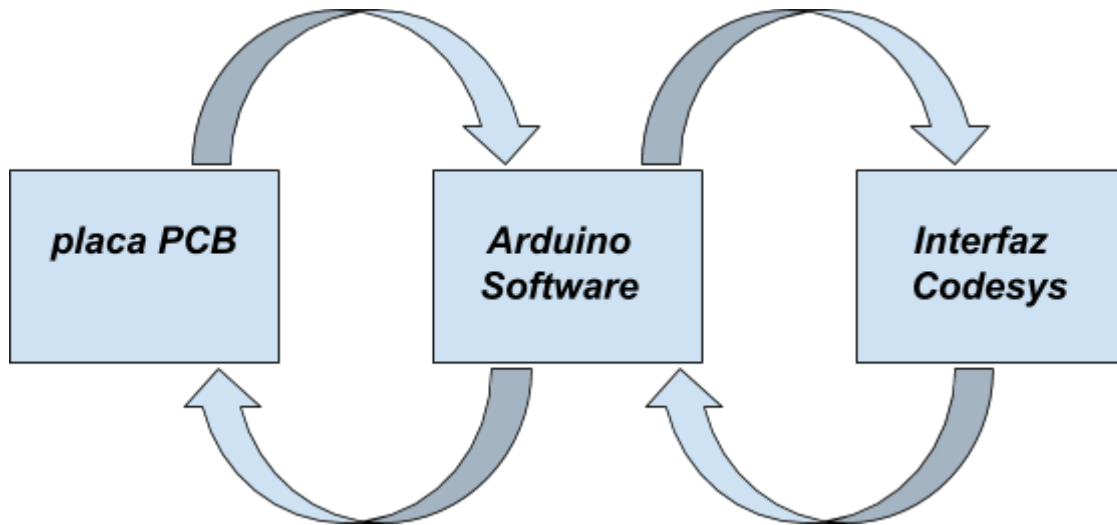
En los ANEXOS se adjunta la hoja de datos para su justificación.



Ilustración 21: Válvula SIRAI de 24V

## 1.9 RESULTADOS FINALES

El funcionamiento de todo el sistema se rige por el siguiente esquema:



Esquema 1: Funcionamiento general del sistema

### 1. Funcionamiento general

El sistema trabaja de la siguiente manera una vez instalado tanto el software como el hardware. La placa lee datos mediante los sensores incorporados, enviando estos valores al programa arduino y este envía dichos valores a la interfaz para que el usuario los visualice y el PID los utilice. A su vez el usuario escribe los valores de presión y caudal que valore que necesita la placa, mediante el PID calcula un valor de referencia que envía al programa arduino y este a su vez lo envía a la placa para que el regulador que incorpora un PID tenga como finalidad llegar al valor deseado.

En resumidas cuentas, la pila de combustible de hidrógeno funciona con hidrógeno y oxígeno con un determinado caudal y presión. El usuario puede controlar ambos valores para su estudio, con la ayuda de los valores medidos reales por los sensores más el valor deseado el controlador automáticamente calcula los valores para hacer esto posible y se encarga de administrar estos datos entre los programas y la placa para su correcto funcionamiento.

### 2. Secciones del sistema

El sistema se puede dividir en 3 apartados para explicar el uso de la placa.

- 1) Válvulas de apertura y cierre: Se activan con Marcha en el programa codesys que activa la etapa de funcionamiento en el arduino, haciendo que se activen las válvulas de apertura y cierre. El arduino a su vez envía información a la interfaz del codesys para indicar que dichas válvulas están en funcionamiento, indicándose con los LEDs de control.
- 2) Válvulas de purga: dichas válvulas, iguales a las anteriores pero con un control más complejo. Por una parte periódicamente la placa arduino recoge los datos de purga escritos por el ingeniero, enviándolos a la placa y al estar en la etapa marcha, el arduino realizará purgas periódicas, enviando la situación de dichas válvulas a la placa para ser visualizado por la interfaz. Pero puede ser que el propio operario desee realizar una purga manual, en ese caso en la botonera de la interfaz puede realizarla, lo que hará será una purga en ese momento con el tiempo de purga que haya seleccionado y reiniciará el período.
- 3) Regulador másico: Dicha parte funciona de la siguiente manera. Por una parte la placa arduino recoge valores de presión y caudal, que son enviados mediante el intercambio de información entre placa y pc al codesys. Por la otra la interfaz recoge el valor que se desea de presión y caudal. Con estos valores, y puestos los valores de  $K_p$ ,  $D$  e  $I$  en los PIDs, constantes necesarias para el funcionamiento del PID, dicho proceso calculará el valor óptimo para ese momento y será enviado a la placa para que actúe con ese valor y así con el tiempo lograr conseguir el valor deseado de presión y caudal.

### *3. Interfaz*

La interfaz está formada por 5 secciones:

- Esquema general del sistema con 4 LED de señalización donde se visualiza la actuación de las válvulas. Indican con color verde o rojo apagado si funcionan, tanto las de apertura y cierre indicadas con LEDs más grandes o las de purga con LEDs menores.
- Botonera: con 4 botones, uno para darle a marcha y que funcione el sistema. dos para realizar tanto una purga manual de oxígeno como de hidrógeno y la palanca de emergencia, que pondrá al sistema en estado de emergencia.
- Gráficas actuación regulador, donde se recoge en tiempo real los valores de referencia que está enviando el sistema al regulador para que este actúe.
- Tablas de valores: hay dos tablas donde el usuario debe rellenarlas con los valores que desee de caudal y presión y el sistema mediante cálculos con el PID llegará a ellos y en la segunda tabla, los valores de período de purga y el tiempo que se desea que duren.

- Por último, las gráficas en tiempo real de los valores de presión y caudal, tanto del hidrógeno como del oxígeno, donde recogen los 3 valores más importantes que son el valor actual medido por los sensores, el valor deseado por el usuario, también llamada referencia y el valor que calcula el PID, es decir la referencia enviada al regulador. Estas gráficas son descargables para estudiar los datos y realizar posibles mejoras.

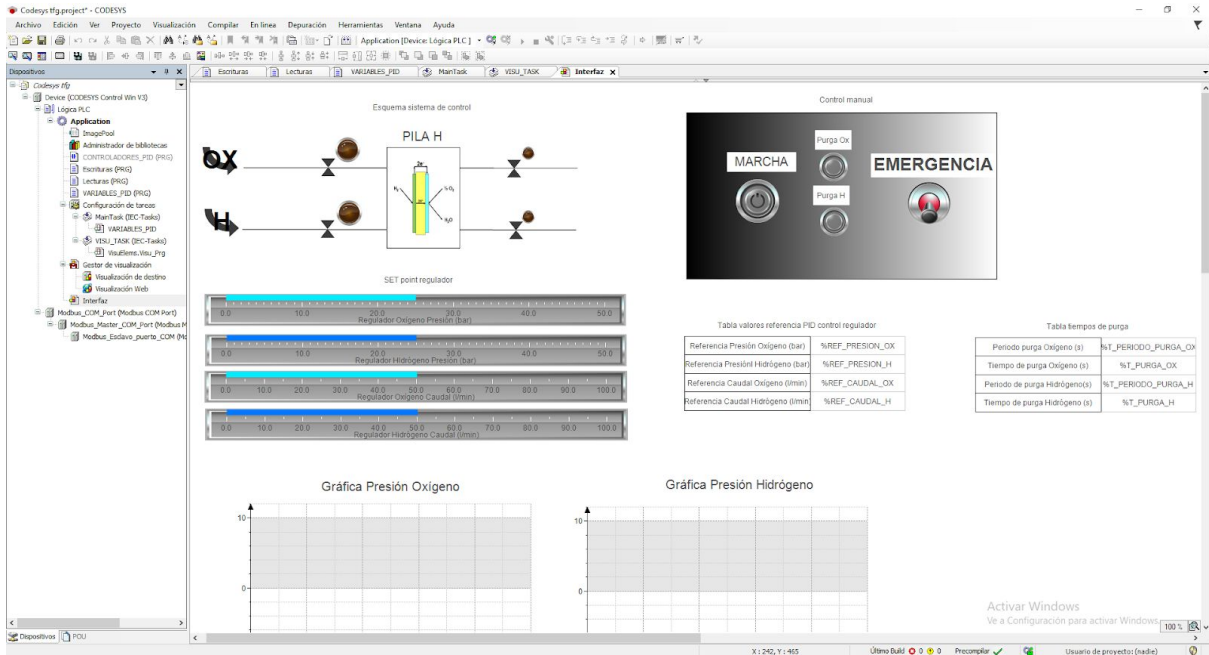


Ilustración 22: Captura de pantalla, interfaz 1 del programa

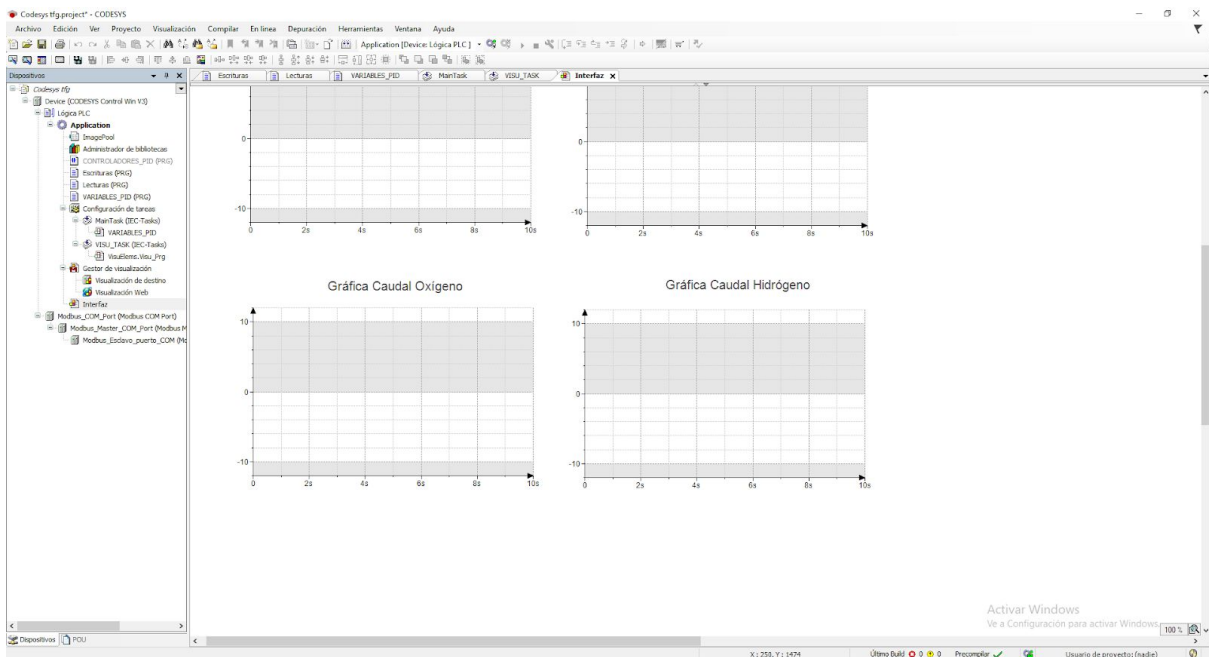


Ilustración 23: Captura de pantalla, interfaz 2 del programa

#### 4. Información en el TFG

La PCB diseñada se encuentra en la sección de PLANOS, con la explicación de los PIN y bornes en el ANEXO.

Los códigos de arduino y codesys se encuentran en los ANEXOS, con su detallada explicación.

Las instrucciones de utilización, que hace referencia a su vez a otras partes del proyecto se encuentran en los ANEXOS.

#### 1.10 PLANIFICACIÓN

El proyecto tenía que desarrollarse durante las horas asignadas para la estancia en prácticas de la asignatura EE 1045, trabajo de final de grado. Siendo de horario libre, el proyectista ha decidido distribuir las 150 horas pertenecientes a las prácticas de la siguiente forma:

	<b>Diseño PCB</b>	<b>Programación Arduino</b>	<b>Programación Codesys</b>
<b>Día 1</b>	6 horas		
<b>Día 2</b>	6 horas		
<b>Día 3</b>	6 horas		
<b>Día 4</b>	6 horas		
<b>Día 5</b>	6 horas		
<b>Día 6</b>	6 horas		
<b>Día 7</b>	6 horas		
<b>Día 8</b>	6 horas		
<b>Día 9</b>	6 horas		
<b>Día 10</b>	6 horas		
<b>Día 11</b>	6 horas		
<b>Día 12</b>	6 horas		

<b>Día 13</b>	6 horas		
<b>Día 14</b>	4 horas	2 horas	
<b>Día 15</b>		6 horas	
<b>Día 16</b>	2 horas	4 horas	
<b>Día 17</b>		6 horas	
<b>Día 18</b>		6 horas	
<b>Día 19</b>		3 horas	3 horas
<b>Día 20</b>			6 horas
<b>Día 21</b>			6 horas
<b>Día 22</b>			6 horas
<b>Día 23</b>			6 horas
<b>Día 24</b>		3 horas	3 horas
<b>Día 25</b>		3 horas	3 horas

Tabla 1: Planificación en horas de las tareas

### 1.11 ORDEN DE PRIORIDAD ENTRE LOS DOCUMENTOS

Este proyecto sigue la norma UNE157001, lo que quiere decir que no modifica el orden de prioridad de los documentos que constituyen del proyecto técnico, siendo el siguiente:

1. Planos.
2. Pliego de condiciones.
3. Presupuesto.
4. Memoria.





# **ANEXOS**



## 2.1 CONTENIDO

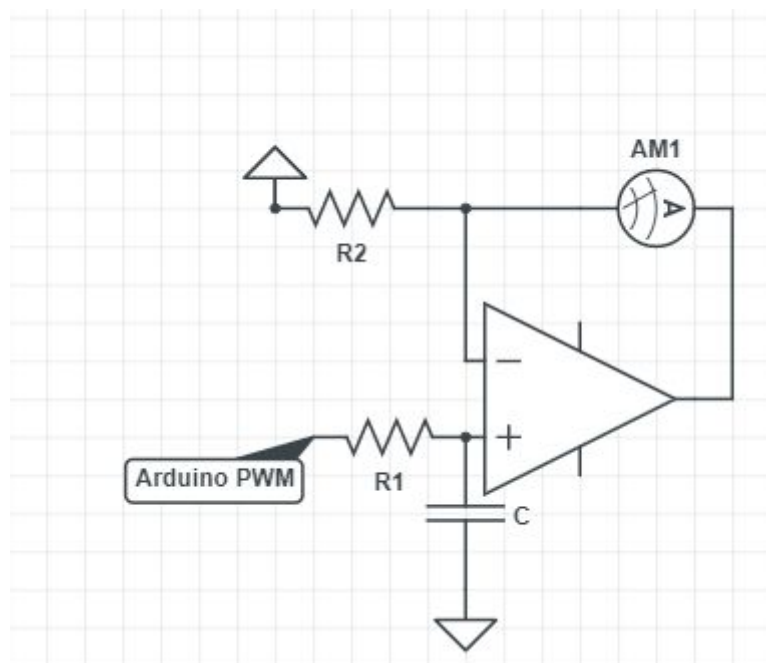
### 2.1.1 CÁLCULOS

En este proyecto se han realizado los siguientes cálculos teniendo en cuenta las consideraciones anteriores:

#### *Cálculo del regulador*

El regulador se conforma con el siguiente circuito, un filtro de paso bajo para convertir la señal PWM del arduino en señal analógica que varía entre 4mA y 20mA.

Para ello se usará un filtro de paso bajo conectado a un amplificador operacional para aprovechar la tensión continua que mantendrá el condensador, con un factor de rizado del 1%, y se tomará como frecuencia de corte 490 Hz, siendo la frecuencia de la PWM 500 Hz.



Esquema 2: Circuito PWM-Continua 4-20mA

Las dos condiciones son transformar la señal PWM en señal continua y que su salida máxima sean 20mA.

$$Uc(t) = Us(\infty) - [ Us(\infty) - Us(0) ] * e^{-\frac{t}{RC}}$$

Siendo 0V y 3,3V el voltaje cuadrado de la PWM. El tiempo es igual a 1/490 segundos y la R1 utilizada será de 1kΩ, queda como resultado despejando la C de 1μF.

Para la R2 se sabe que el voltaje será 3.3V en su máximo, y la corriente será de 20mA en ese caso, por tanto aplicando la ley de ohm la R2 es de 165Ω.

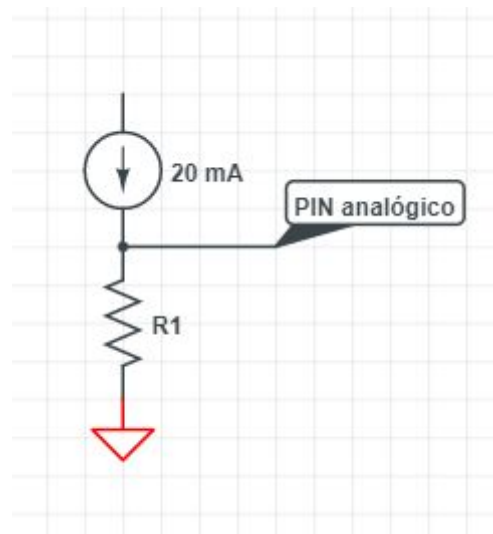
$$V = R * I$$

### *Cálculo de la lectura de sensores presion y caudal*

El arduino due puede medir mediante sus pin analógicos señales entre 0 y 3.3V y dar un número de bits entre 0 y 1024 según la medida.

Los sensores pueden dar una señal de 4-20mA o bien una señal de 1-5V. Para conseguir que el valor máximo medido por la tarjeta sean de 3.3V se realiza un divisor de tensión.

### *Caso 1: 4-20mA*

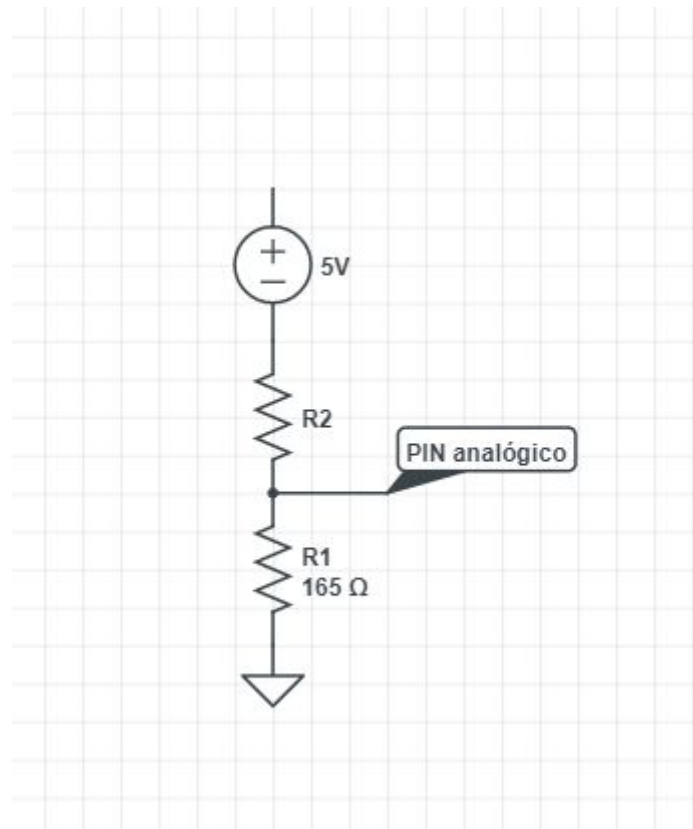


Esquema 3: Circuito lectura sensores 4-20mA

Con un divisor de tensión, sabiendo que el PIN analógico solo puede medir 3.3 V como máximo y conociendo la corriente, aplicando la ley de ohm se tiene el valor de la R1, que es: 165 Ω.

$$V = R * I$$

Caso 2: 1-5V

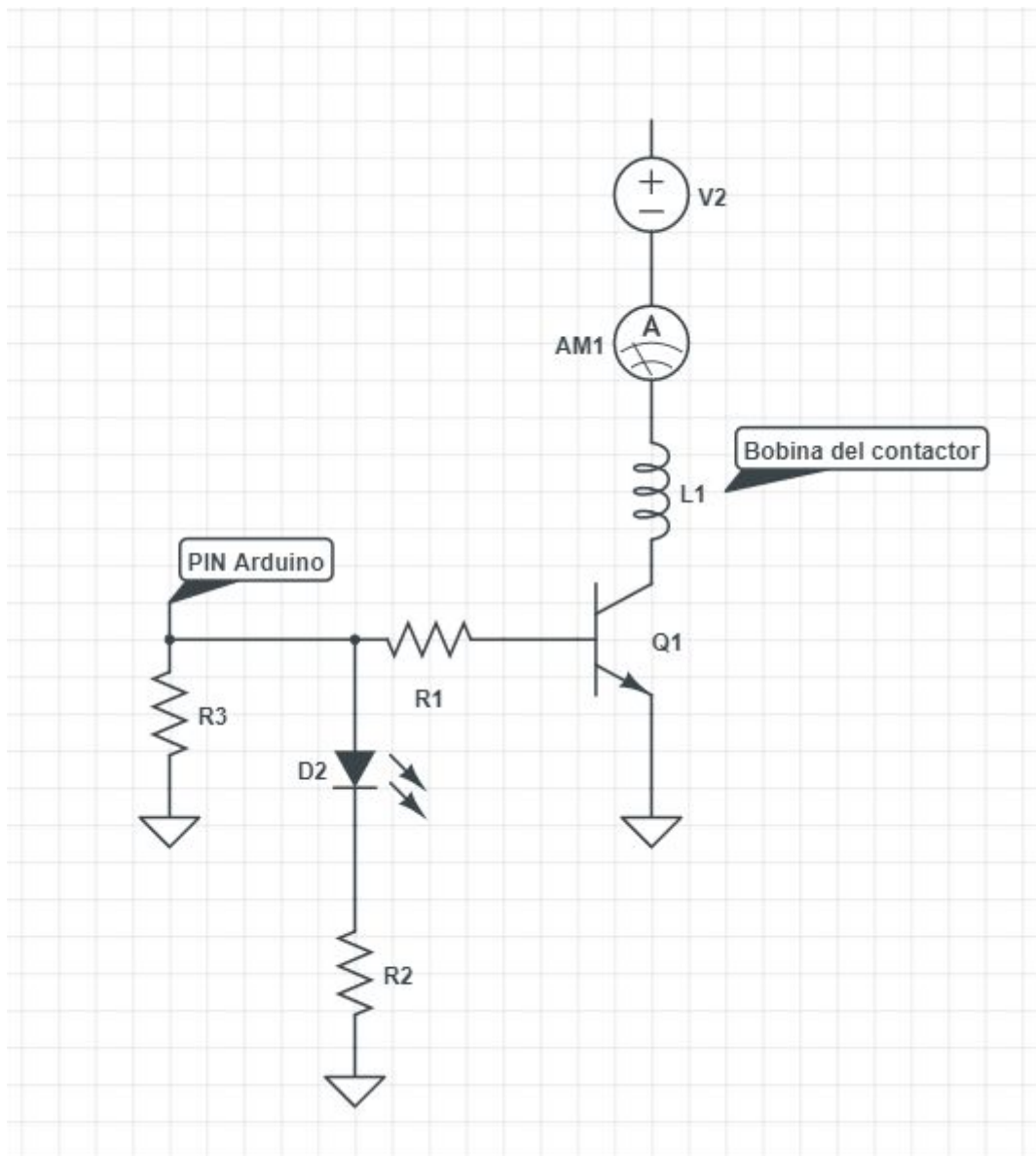


Esquema 4: Circuito lectura sensores 4-20mA y 0-5V

Teniendo en cuenta que se debe mantener la resistencia anterior, ahora también se realiza un divisor de tensión pero esta vez para reducir los 5V que como máximo proporcionará el sensor a 3.3V. Para ello con la ley de ohm sabiendo que la intensidad máxima que recorrerá el circuito seguirá siendo 20 mA y la diferencia de tensión para que eso se cumpla en la resistencia, será de  $5-3.3V=1,7V$ . Por tanto aplicando la ley de ohm se obtiene una R2 de  $85\Omega$ .

$$V = R * I$$

### Cálculo del circuito del contactor



Esquema 5: Circuito actuación sobre contactores

Con el contactor elegido se tiene que la bobina tiene una resistencia de  $69,4\Omega$  por su hoja de datos. Sabiendo que se activa con  $5V$ , es necesario que recorra esa rama del circuito  $0,07A$ , aplicando la ley de ohm.

$$V = R * I$$

En la base del colector sabiendo esa intensidad, podemos saberla aplicando los datos conocidos que son la beta del transistor, siendo 10.  $I_b = 7,2 \text{ mA}$ .

$$I_c = I_b *$$

Sabiendo que la tensión Vb-e es igual a 0,7 por la hoja de datos, aplicando la ley de ohm se sabe que la resistencia para conseguir dicha intensidad de 7,2 mA, el voltaje que habrá será de  $3,3 - 0,7 = 2,6V$ . Entonces se tiene una resistencia de aproximadamente  $361\Omega$  que se redondea a  $300\Omega$  para ir más holgados en intensidad.

$$V = R * I$$

Por otra parte, la R2 necesaria para encender el led utilizando el mínimo de intensidad es de 4mA con la hoja de datos. El voltaje necesario para que el led entre en conducción es de 2,1V, por tanto  $3,3 - 2,1$  son 1,2V. Aplicando la ley de ohm de nuevo tenemos una resistencia de  $300\Omega$ .

$$V = R * I$$

### *Corriente máxima placa arduino*

La placa arduino tiene un gran número de pin con los que se pueden elaborar funciones muy útiles pero tiene un límite en cuanto a intensidad puede suministrar a los componentes. Se va a suponer en este cálculo que todos los dispositivos a los que la placa entrega potencia, están conectados y funcionando al máximo. No debe de superar dicho valor los 130mA sin contar los PIN de 5V y 3.3V. Se utilizará un coeficiente de seguridad de 1,5. Como se observa en la siguiente tabla, se cumple y nunca se superará la máxima corriente suministrada por la placa.

<b>Circuito</b>	<b>Consumo</b>	<b>Unidades</b>	<b>Total</b>
Regulador	3,4mA	4	13,6 mA
Contactador	12mA	4	48 mA
LED	4mA	4	16 mA
<b>TOTAL (multiplicado por 1,5)</b>			116,4 mA

Tabla 2: Estudio consumo placa arduino Due

### 2.1.2 ESTUDIO BÁSICO DE SEGURIDAD Y SALUD

Este proyecto no contendrá un estudio de seguridad y salud por las siguientes razones, en su lugar será redactado un estudio Básico de seguridad y salud.

a) Que el presupuesto de ejecución por contrata incluido en el proyecto sea igual o superior a 75 millones de pesetas.

b) Que la duración estimada sea superior a 30 días laborables, empleándose en algún momento a más de 20 trabajadores simultáneamente.

c) Que el volumen de mano de obra estimada, entendiendo por tal la suma de los días de trabajo del total de los trabajadores en la obra, sea superior a 500.

d) Las obras de túneles, galerías, conducciones subterráneas y presas.

En este proyecto no se describen las obras a realizar, por tanto no se pueden explicar las medidas de seguridad para llevarlo a cabo.

Una vez el proyecto esté completo, consta de dos salas, una con el PC donde se situará el/los científicos y otra donde irá la pila de hidrógeno con el combustible que necesita más los componentes descritos en el proyecto para su correcto funcionamiento, sumado a los sensores que sean necesarios para el estudio eléctrico de ésta. Esta sala tendrá incorporada un extractor de gas industrial funcionando en todo momento que la prueba dure por posibles fugas y una seta de emergencia.

Teniendo en cuenta lo anterior, las medidas a tomar son las siguientes:

-Asegurar que no existe ninguna fuga de gases, ya que el hidrógeno es explosivo y el oxígeno altamente inflamable.

-En funcionamiento debe siempre estar encendido el extractor de gas.

-Colocar la seta de emergencia cerca de la zona de trabajo, en el caso de que se tuviera que entrar para cambiar la pila u otro componente, que fuera sencillo la pulsación de esta.

-Prohibido fumar en ambas salas.

-Prohibido cualquier material inflamable, con alta variabilidad térmica o propenso a generar chispas en ambas salas sin contar lo absolutamente necesario como PC y componentes.

-Ante cualquier indicio de flama, apretar el pulsador de emergencia.

Los riesgos imposibles de evitar son una explosión o algo similar, ya que al trabajar con Oxígeno e Hidrógeno puro es imposible, solo minimizable. Por ello se han redactado las indicaciones anteriores, se le suma un protocolo cuando se activa el estado de emergencia para minimizar el riesgo de fuga y por ende de explosión.

En caso de fallo catastrófico, seguir las indicaciones del edificio de la CEU en el que está ubicado el laboratorio. Abandonar el edificio por el camino de emergencia más corto que indican los paneles indicativos.



### 2.1.3 HOJAS DE DATOS

## Process Gas Mass Flow Meters and Controllers

### FEATURES

- Direct monitoring of mass flow eliminates need for ancillary pressure and temperature sensing
- Patented straight sensor tube with access ports permits easy cleaning and reduces maintenance down-time
- Platinum sensor eliminates zero-drift and ensures long-term repeatability
- Fast-response control valve provides rapid response to set point changes and operates over a wide pressure differential range
- Primary standard calibration ensures starting point accuracy and NIST traceability
- Available with a wide variety of enclosures, process connections, input/output options and control electronics
- CE Approved

# SideTrak® 830/840



### DESCRIPTION

Sierra Instruments' Side-Trak® Model 830 Mass Flow Meters and Model 840 Mass Flow Controllers are designed for precise measurement and control of air and process gases in ranges from 0 to 10 sccm up to 0 to 500 slpm. Because all wetted materials are 316 stainless steel, the device accommodates most clean gases, including corrosives.

Proven by over 25 years of field installations, the Side-Trak product line is distinguished by its patented, cleanable, large-diameter sensor tube and the reliability and serviceability that this feature provides. Because a microscopic layer of contamination has a major effect on small diameter sensor tubes, many mass flow meters and controllers suffer, over time, from degradation of accuracy or repeatability.

The Side-Trak sensor is not only larger in diameter than most other MFMs and MFCs, but it is mounted along the side of the flow body to provide access ports at either end. In the event of clogging or contamination, it can be cleaned with the 0.03-inch diameter cleaning rod available from Sierra.



[www.sierrainstruments.com](http://www.sierrainstruments.com)



## DESCRIPTION (continued)

Sierra's SideTrak® sensor also overcomes the problem of zero-drift commonly associated with capillary-type meters and controllers. Because the sensor windings are constructed of platinum, the NIST standard for temperature detection and one of the most stable elements known, the device is virtually drift-free.

Sierra's Model 840 features a built-in electromagnetic servo-control valve that provides precise, instantaneous control of gas delivery to a test, batch or continuous process operation. Speed-of-response, accuracy and reliability characterize the Model 840 and have made it the instrument of choice in a wide variety of gas flow control applications—from laboratory and test benches to instrument OEMs, in analytical and process industries, and as a transfer standard in metrology labs.

SideTrak's broad range of sizes, control electronics, process connections, enclosures, input/output options, cables and connectors provide flexibility, versatility—ultimately, the ideal instrument package for your specific application.

### LOW FLOW BODY

FLOW RANGE	
0-10 sccm to 0-15 slpm	
Process Tubing	.25 (6.4)

FITTING TYPE, .562-18 THREAD <sup>(1)</sup>			
	Compression .125 or .25	VCO (male) .25	VCR (male) .25
Dim. L	4.84 or 5.0	4.60	4.90

### MEDIUM FLOW BODY

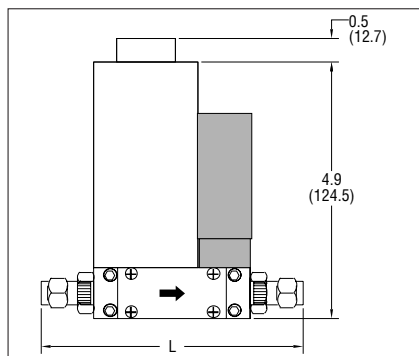
FLOW RANGE, SLPM				
	0-15	0-30	0-50, 100	0-100
Process Tubing	.25 or .375 (6.4 or 9.5)	.25 or .375 (6.4 or 9.5)	.25 or .375 (6.4 or 9.5)	.375 (9.5)

FITTING TYPE, .562-18 THREAD <sup>(1)</sup>			
	Compression .25 or .375	VCO (male) .25 or .375	VCR (male) .25 or .375
Dim L.	6.27 or 6.39	5.81 or 6.25	.613 or 6.25

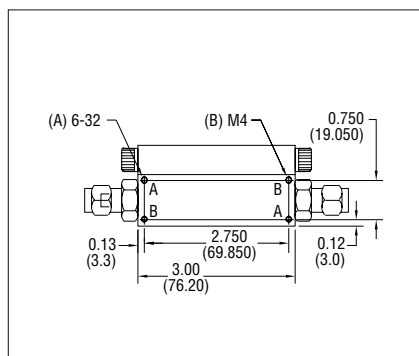
Note: (1) Metric fittings are available, consult factory.

All dimensions are inches. Millimeters are in parentheses. Certified drawings are available on request.

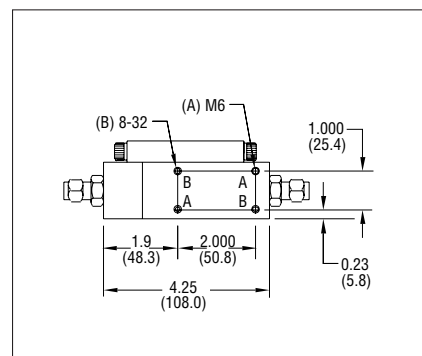
Model 830L/840L — Front View



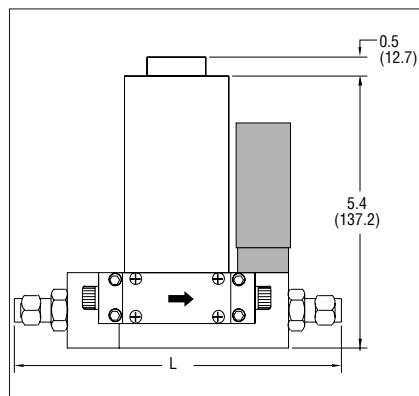
Model 830L/840L — Bottom View



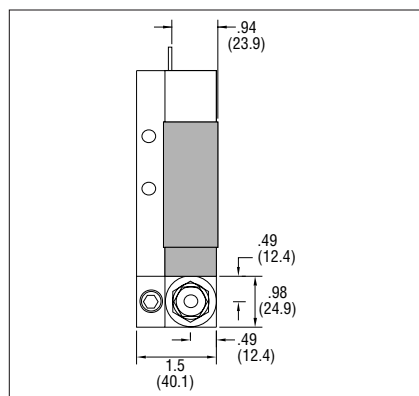
Model 830M/840M — Bottom View



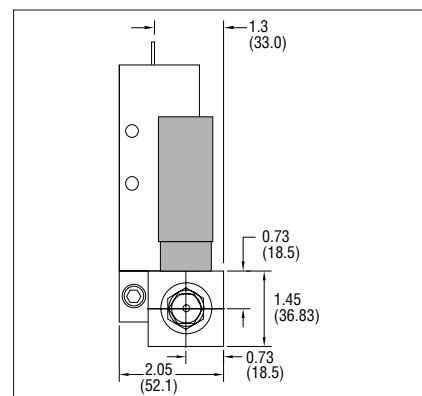
Model 830M/840M — Front View



Model 830L/840L — Outlet View



Model 830M/840M — Outlet View



Shaded portion of dimensional drawings indicates Model 840 controllers.

**HIGH FLOW BODY**

**FLOW RANGE, SLPM**

	0-100	0-200	0-300	0-400	0-500
<b>Process Tubing</b>	.375 or .50 (9.5 or 12.7)	.375 or .50 (9.5 or 12.7)	.50 (12.7)	.50 (12.7)	.50 (12.7)

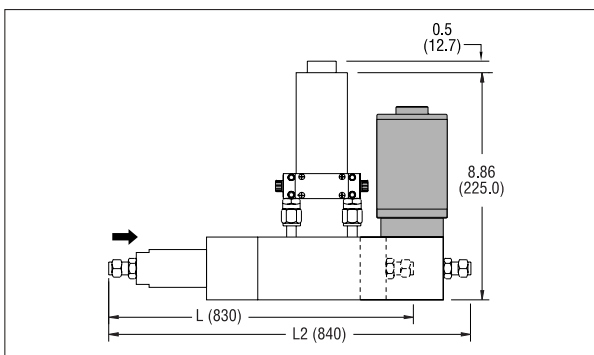
**FITTING TYPE, .75-16 THREAD <sup>(1)</sup>**

	Compression .375 or .50	VCO (male) .375	VCR (male) .375
<b>Dim. L 830</b>	11.90 or 12.10 (302.3 or 307.3)	11.80 (299.7)	12.19 (309.6)
<b>Dim. L2 840</b>	14.10 or 14.30 (358.1 or 363.2)	14.00 (355.6)	14.40 (365.8)

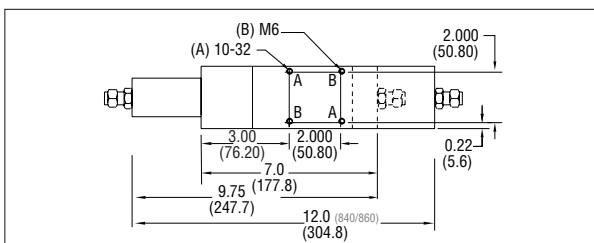
Note: (1) Metric fittings are available, consult factory.

All dimensions are inches. Millimeters are in parentheses. Certified drawings are available on request.

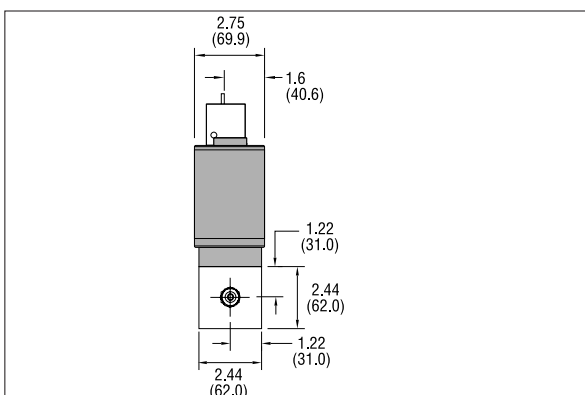
**Model 830H/840H — Front View**



**Model 830H/840H — Bottom View**



**Model 830H/840H — Outlet View**



Shaded portion of dimensional drawings indicates Model 840 controllers.

**PERFORMANCE SPECIFICATIONS**

**Accuracy**

+/- 1% of Full Scale under calibration conditions including linearity over 15 to 25°C and 10 to 60 psia (0.7 to 4 bara); +/- 2% of Full Scale including linearity under calibration conditions over 0 to 50°C and 5 to 150 psia (0.3 to 10 bara); +/- 1% of Full Scale accuracy at a specific temperature and pressure is available with special calibration.

**OPERATING PRESSURE <sup>(1)</sup>**

Inlet Pressure Deviation <sup>(1)</sup>	50 psig	100 psig	150 psig
+/- 1 psig	+/- 1% of full scale	+/- 1% of full scale	+/- 1% of full scale
+/- 5 psig	+/- 2.5% of full scale	+/- 3% of full scale	+/- 3.5% of full scale
+/- 10 psig	+/- 4% of full scale	+/- 5% of full scale	+/- 6% of full scale

Notes: (1) Do not exceed 150 psig.

(2) Difference between inlet pressure and calibrated pressure. Do not exceed +/- 10 psig.

**Repeatability**

+/- 0.15% of Full Scale  
+/- 0.5% of Full Scale for 840 (NPT)

**Temperature Coefficient**

0.08% of Full Scale per °F (0.15% of Full Scale per °C), or better

**Pressure Coefficient**

0.01% of Full Scale per psi (0.15% of Full Scale per bar), or better

**Response Time<sup>(1)</sup>**

- 830 (all) . . . . . 300 ms time constant; 2 second (typical) to within +/-2% of set point over 20 to 100% of Full Scale
- 840 (low, med) . . . . . 300 ms time constant; 2 second (typical) to within +/-2% of set point over 20 to 100% of Full Scale
- 840 (high) . . . . . 600 ms time constant; 4 seconds (typical) to within +/-2% of set point over 20 to 100% of Full Scale

Notes: Option available on 840 low, medium and high flow bodies of 1.5 second time constant, others on special order.

## OPERATING SPECIFICATIONS

### Gases

All clean gases; specify when ordering

### Mass Flow Rate

0 to 10 sccm to 0 to 500 slpm; flow ranges specified are for an equivalent flow of nitrogen at 760 mm Hg and 21°C (70°F); other ranges in other units are available (e.g., scfh or nm<sup>3</sup>/h)

### Gas & Ambient Temperature

32° to 176°F (0 to 80°C); above 122°F (50°C) requires HT or RT option.

### Gas Pressure

500 psig (34 barg) maximum;  
30 psig (2 barg) optimum

### Leak Integrity

5 X 10<sup>-9</sup> atm cc/sec of helium maximum

### Power Requirements

830 (all) +15 VDC @ 80 mA, 1.2 watts and  
-15 VDC @ 10 mA, 0.15 watts  
840 (low) +15 VDC @ 130 mA, 2 watts and  
-15 VDC @ 200 mA, 3.0 watts  
840 (med) +15 VDC @ 400 mA, 6 watts and  
-15 VDC @ 300 mA, 4.5 watts  
840 (high) +15 VDC @ 1.13 A, 17 watts and  
-15 VDC @ 1.06 A, 16 watts

### Control Range

840 (all) 2 to 100% of Full Scale;  
valve shuts between 1.0% to 2.5% of Full Scale

### Command Signal (For 840 Only)

0 to 5 VDC, 20 megaohms minimum input impedance  
4 to 20 mA, 250 ohms maximum input impedance

### Output Signal

Linear 0 to 5 VDC, 1000 ohms minimum load resistance  
Linear 4 to 20mA, 500 ohms maximum loop resistance

## PHYSICAL SPECIFICATIONS

### Wetted Materials

830 (all) 316 stainless steel, Viton® "O"-rings standard;  
Neoprene and 4079 Kalrez® optional;  
others on special order  
840 (low, med, high) 316 stainless steel, 430F stainless steel,  
Viton® "O"-rings and valve seat standard;  
Neoprene, 4079 Kalrez® and other elastomers available on special  
order;  
PFA Teflon® valve seat available

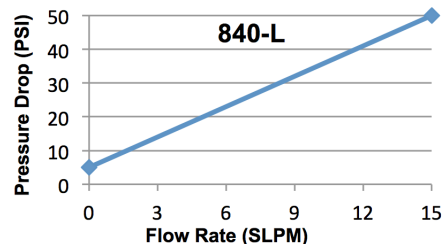
### Control Valve Type

840 (low, med, high) Electromagnetic

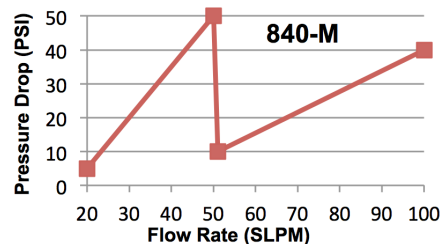
### Differential Pressure Requirements $\Delta P$ , For Model 840

30 psi (2.1 bar) differential optimum for all controllers

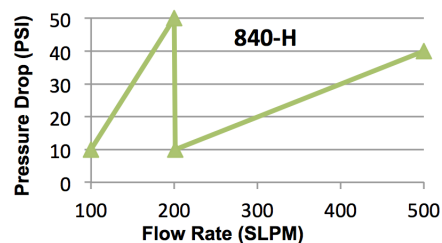
840 (low) .010 - 15 slpm:  
5 to 50 psi (0.3 to 3.4 bar) differential standard;



840 (med) 15 to 50 slpm:  
5 to 50 psi (0.3 to 3.4 bar) differential;  
51 to 100 slpm:  
10 to 40 psi (0.7 to 2.8 bar) differential standard;



840 (high) 100 to 200 slpm:  
10 to 50 psi (0.7 to 3.4 bar) differential;  
201 to 500 slpm:  
10 to 40 psi (0.7 to 2.8 bar) differential standard;  
lower or higher  $\Delta P$  available on special order



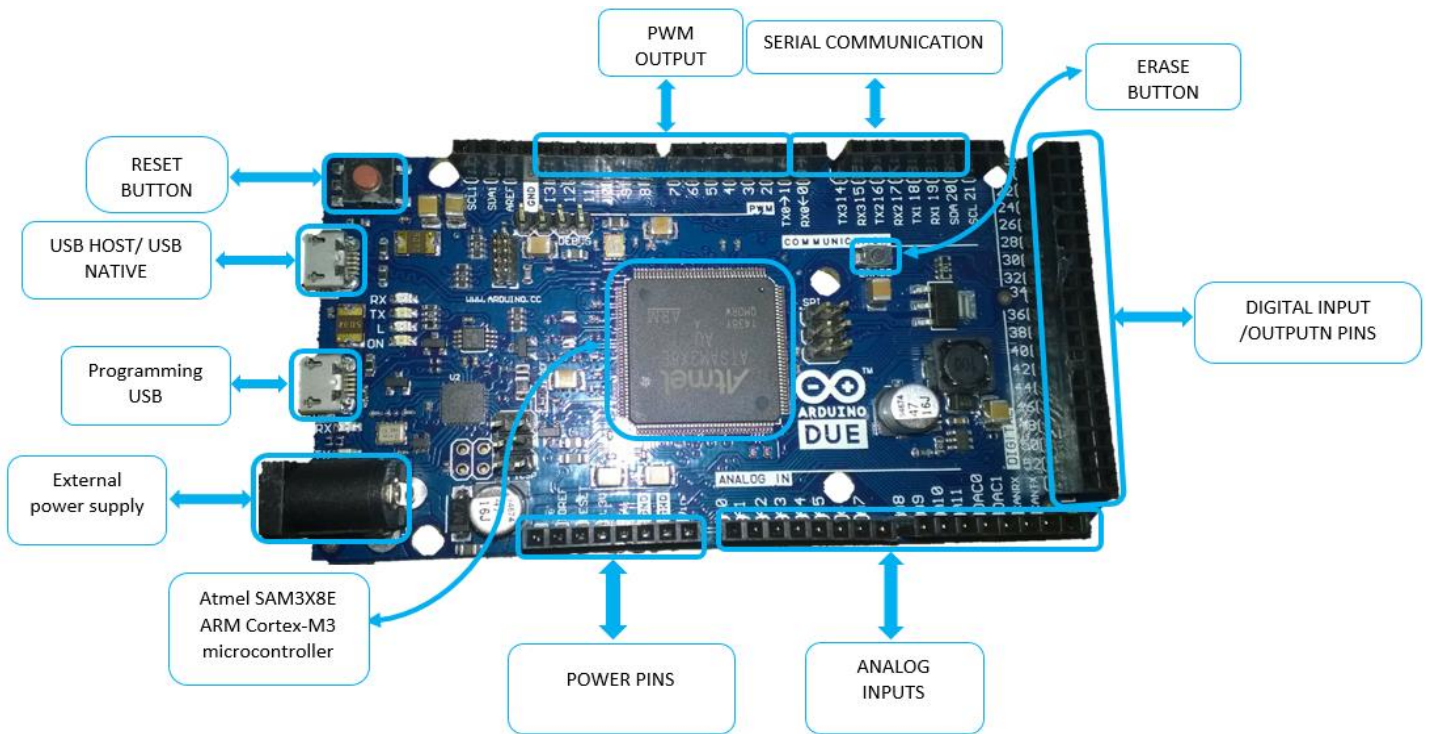
### PRESSURE DROP ACROSS THE 830

Flow Meter	Flow rate	psi			Mbar		
		1/4	3/8	1/2	1/4	3/8	1/2
Meter Fittings (inch)		1/4	3/8	1/2	1/4	3/8	1/2
830-L	100 sccm	.005	n/a	n/a	.309	n/a	n/a
830-L	500 sccm	.006	n/a	n/a	.440	n/a	n/a
830-L	1 slpm	.006	n/a	n/a	.475	n/a	n/a
830-L	15 slpm	.11	.06	n/a	7.59	4.0	n/a
830-M	20 slpm	.08	.03	n/a	5.50	2.0	n/a
830-M	50 slpm	n/a	.3	.10	n/a	20.4	6.8
830-M	100 slpm	2.0	.85	.30	136	58	20.4
830-H	200 slpm	n/a	2.50	2.00	n/a	170	136
830-H	500 slpm	n/a	n/a	n/a	n/a	n/a	578

		830											
<b>PARENT NUMBER</b>													
<b>830</b>	SideTrak® Mass Flow Meter												
<b>840</b>	SideTrak® Mass Flow Controller												
<b>FLOW BODY</b>													
<b>L</b>	Low Flows (0–10 sccm to 0–15 slpm)												
<b>M</b>	Medium Flows (0–15 slpm to 0–100 slpm)												
<b>H</b>	High Flows (0–100 slpm to 0–500 slpm)												
<b>INLET/OUTLET FITTINGS</b>													
<b>7</b>	1/2-inch VCO (High Flow Only)												
<b>1</b>	1/8-inch Compression					<b>8</b>	1/4-inch VCR						
<b>2</b>	1/4-inch Compression					<b>9</b>	1/2-inch VCR						
<b>3</b>	3/8-inch Compression					<b>10</b>	6 mm Compression						
<b>4</b>	1/2-inch Compression					<b>11</b>	10 mm Compression						
<b>5</b>	1/4-inch VCO					<b>12</b>	12 mm Compression						
<b>6</b>	1/2-inch VCO					<b>13</b>	1/4-inch NPT, female						
<b>FLOW BODY ELASTOMERS</b>													
<b>OV1</b>	Viton® (Low, Medium, High Flow)												
<b>ON1</b>	Neoprene (Low, Medium Flow)												
<b>ON2</b>	Neoprene (High Flow)												
<b>OK1</b>	Kalrez® (Low Flow)												
<b>OK2</b>	Kalrez® (Medium Flow)												
<b>OK3</b>	Kalrez® (High Flow-830)												
<b>OK4</b>	Kalrez® (High Flow-840)												
<b>VALVE SEAT (840, 840-CE ONLY)</b>													
<b>SV1</b>	Viton® (Standard)												
<b>SN1</b>	Neoprene or Equivalent												
<b>SK1</b>	Kalrez® or Equivalent (Low, Medium Flow)												
<b>SK2</b>	Kalrez® or Equivalent (High Flow)												
<b>ST1</b>	PFA Teflon or Equivalent												
<b>ELECTRONIC CONNECTORS</b>													
<b>E</b>	Edge-Card Connector												
<b>D</b>	15 Pin "D" Connector												
<b>OUTPUT SIGNAL</b>													
<b>V1</b>	0 to 5 VDC, Linear												
<b>V4</b>	4 to 20 mA, Linear												
<b>COMMAND SIGNAL (840 ONLY)</b>													
<b>S1</b>	0 to 5 VDC												
<b>S4</b>	4 to 20 mA												
<b>OPTIONS 1</b>													
<b>HT</b>	High Temperature Calibration												
<b>RT(ft)</b>	Remote High Temperature Calibration (Cable Length in Feet)												
<b>MP</b>	Medium Pressure Calibration, 40 to 100 psig (2.8 to 6.9 barg)												
<b>HP</b>	High Pressure Calibration, 100 to 500 psig (6.9 to 34.5 barg)												
<b>LF</b>	Low Flow Calibration, (0 to 20 sccm or below)												
<b>GAS, FLOW RATE</b>													

CONNECTORS, CABLES AND ACCESSORIES (Consult Factory)

## ARDUINO DUE



## INTRODUCTION

The Arduino Due is the first Arduino board based on a 32-bit ARM core microcontroller. It is suitable for large scale projects and it differs from other Arduino kits as it runs at 3.3V as the maximum voltage.

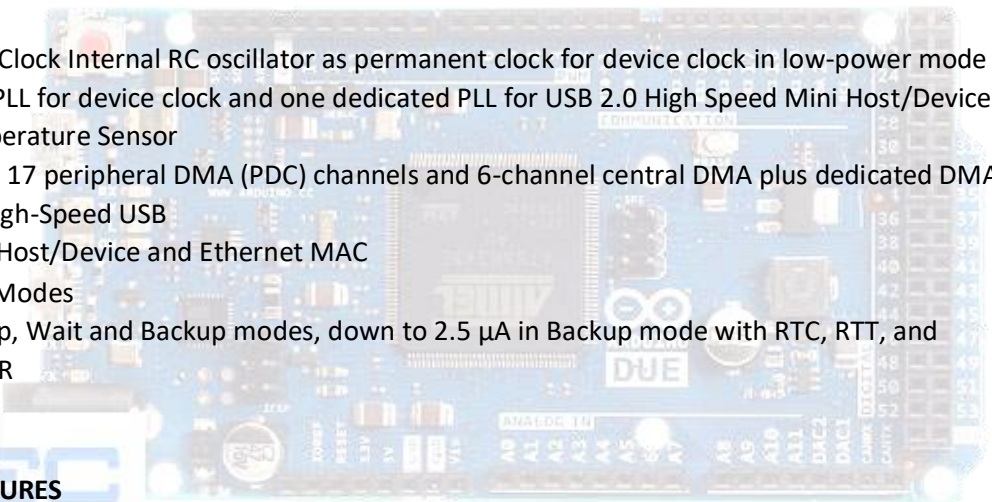
## ARDUINO DUE PHYSICAL COMPONENTS

It is based on **Atmel SAM3X8E ARM Cortex-M3** microcontroller, which is a member of a family of Flash microcontrollers based on the high performance 32-bit ARM Cortex-M3 RISC processor. It operates at a maximum speed of 84 MHz and up to 512 Kbytes of Flash memory and up to 100 Kbytes of SRAM. And its features are as follows:

- Core
  - ARM Cortex-M3 revision 2.0 running at up to 84 MHz
  - Memory Protection Unit (MPU)
  - 24-bit SysTick Counter
  - Nested Vector Interrupt Controller



- Memories
  - 256 to 512 Kbytes embedded Flash, 128-bit wide access, memory accelerator, dual bank
  - 32 to 100 Kbytes embedded SRAM with dual banks
  - 16 Kbytes ROM with embedded bootloader routines (UART, USB) and IAP routines
  - Static Memory Controller (SMC): SRAM, NOR, NAND support. NFC with 4 Kbyte RAM buffer and ECC
  
- System
  - Embedded voltage regulator for single supply operation
  - Power-on-Reset (POR), Brown-out Detector (BOD) and Watchdog for safe reset
  - Quartz or ceramic resonator oscillators: 3 to 20 MHz main and optional low power 32.768 kHz for RTC or device clock
  - High precision 8/12 MHz factory trimmed internal RC oscillator with 4 MHz default frequency for fast device
  
- startup
  - Slow Clock Internal RC oscillator as permanent clock for device clock in low-power mode
  - One PLL for device clock and one dedicated PLL for USB 2.0 High Speed Mini Host/Device
  - Temperature Sensor
  - Up to 17 peripheral DMA (PDC) channels and 6-channel central DMA plus dedicated DMA for High-Speed USB
  - Mini Host/Device and Ethernet MAC
  
- Low-power Modes
  - Sleep, Wait and Backup modes, down to 2.5  $\mu$ A in Backup mode with RTC, RTT, and GPBR

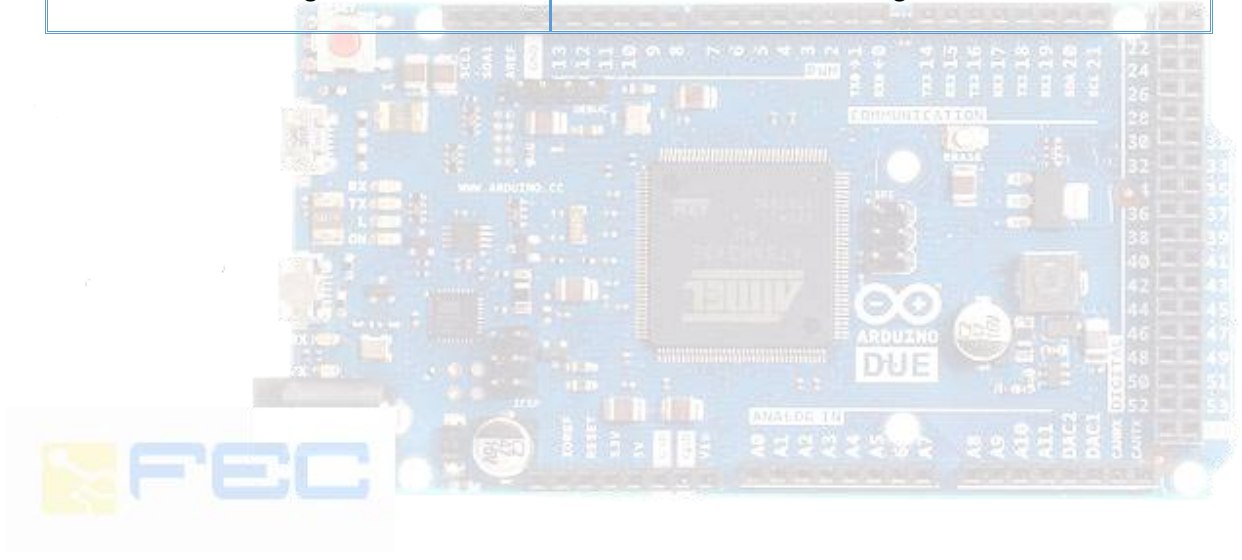


## ARDUINO DUE FEATURES

Microcontroller	AT91SAM3X8E
Operating Voltage	3.3V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-16V
Digital I/O Pins	54 (of which 12 provide PWM output)
Analog Input Pins	12
Analog Output Pins	2 (DAC)



Total DC Output Current on all I/O lines	130 mA
DC Current for 3.3V Pin	800 mA
DC Current for 5V Pin	800 mA
Flash Memory	512 KB all available for the user applications
SRAM	96 KB (two banks: 64KB and 32KB)
Clock Speed	84 MHz
Length	101.52 mm
Width	53.3 mm
Weight	36 g



# Low Pressure Transmitter for Industrial Applications

S I L

## Main features

- Measuring ranges 0...10 mbar to 0...40 bar
- Standard signals 4...20 mA, 0...10 V, 1...5 V
- Highly flexible options by its modular design
- Highly reliable

## Applications

- General industrial use
- Hydraulics
- Pneumatics
- Mechanical engineering
- Plant engineering and automation technology

## Description

The Si-based pressure sensors which in their external design are comparable to the SML model can make use of the advantages of silicon technology. These benefits include lower overall production costs. Thanks to its design, all customary and client-specific pressure connection configurations are possible. Also, the complete range of electrical adapters, which are already known from the SML series, can be integrated.

Its modular design permits reasonable manufacture also in medium-size batches that can be supplied within short periods of time.

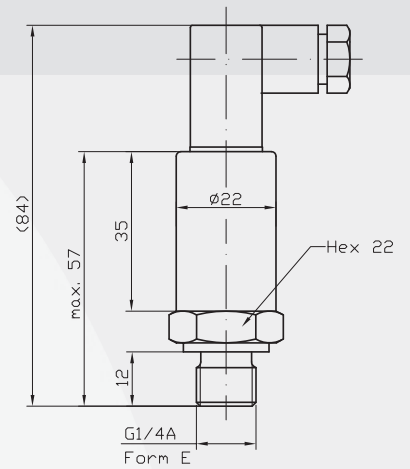
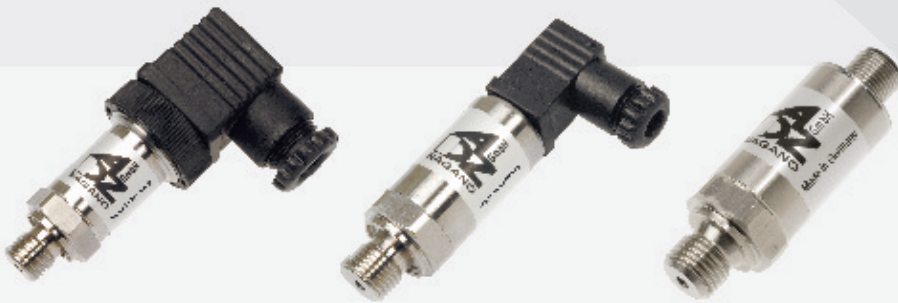


### Specifications

<b>Pressure range</b>								
Measuring range*	p [mbar]	10	16	20	25	40	60	100
Overload pressure	p [mbar]	300	300	300	300	300	300	300
Burst pressure	p [mbar]	500	500	500	500	500	500	500
Measuring range*	p [mbar]	160	200	250	400	600	1000	
Overload pressure	p [mbar]	300	300	2000	2000	2000	2000	
Burst pressure	p [mbar]	500	500	3000	3000	3000	3000	
Measuring range*	p [bar]	1,6	2,0	2,5	4,0	6,0	10,0	
Overload pressure	p [bar]	6	6	6	10	20	20	
Burst pressure	p [bar]	9	9	9	15	30	30	
Measuring range*	p [bar]	16	20	25	40			
Overload pressure	p [bar]	40	40	100	100	(vacuum, relative pressure, + -		
Burst pressure	p [bar]	60	60	150	150	or absolute pressure are available)		
<b>Electrical parameter</b>								
Output signal * and maximum acceptable burden $R_A$	$R_A$ in Ohm	signal			$U_s$ [V <sub>DC</sub> ]	$R_L$ [k $\Omega$ ]	RA [ $\Omega$ ]	
		4...20 mA	(2-wire, 3-wire)		9...32		acc. to $R_A = (U_s - 10V) / 0,02 A$	
		0...10 V <sub>DC</sub>	(3-wire)		12...32	> 5,0		
		1...5 V <sub>DC</sub>			8...32	> 1,0		
		0,5...4,5 V <sub>DC</sub>	ratiometric		5 $\pm$ 10%	> 4,7		
Response time * (10-90%)	t [ms]	< 1						
Withstand voltage	U [V <sub>DC</sub> ]	350						
<b>Accuracy</b>								
Accuracy @RT	% of the range	$\leq 1,0^{**}$	Option $\leq 0,5$		** incl. nonlinearity, hysteresis, repeatability, zero-offset- and final-offset (acc. to IEC 61298-2)			
	BFSL	$\leq 0,25$						
Non-linearity	% of the range	$\leq 0,15$						
Repeatability	% of the range	$\leq 0,10$						
Stability/year	% of the range	$\leq 0,10$						
<b>Acceptable temperature ranges</b>								
Measuring medium	T [°C]	-40...85						
Ambience	T [°C]	-40...85						
Storage	T [°C]	-40...85						
Compensated range*	T [°C]	-10...70						
<b>Temperature coefficient within the compensated range</b>								
Mean TC offset	% of the range	$\leq 0,15 / 10K$						
Mean TC range	% of the range	$\leq 0,15 / 10K$						
Total error	% of the range	-40°C	3,00%					
	% of the range	85°C	3,00%					
<b>Mechanical parameter</b>								
Parts in contact with the measuring medium*		silicon						
Housing*		stainless steel						
Shock resistance	g	1000	acc. to IEC 68-2-32					
Vibration resistance	g	20	acc. to IEC 68-2-6 and IEC 68-2-36					
Mass	m [g]	80-120	depending on design					
CE - conformity		EC Directive 89/336/EWG						
IP system of protection		The IP system of protection as specified in the data sheets generally applies, with their mating plug connected. Relative pressure transmitters usually require a ventilated mating plug and/or cable to allow for pressure compensation.						
* other upon request		From a pressure range of 60bar, a ventilated mating plug and/or cable is not necessarily required.						

Configurations -examples-

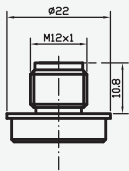
SIL with MVS/C connector



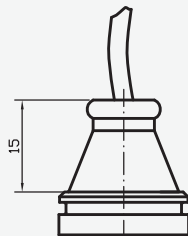
(deviations for absolute pressure are possible)

Connectors\*

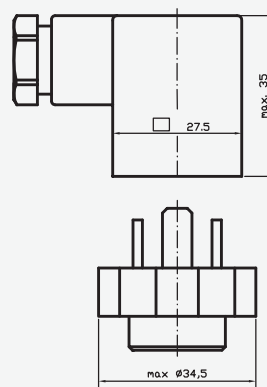
male socket  
M12x1 (S 763)



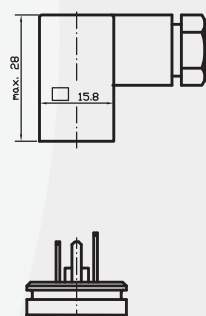
cable output



MVS/A  
DIN EN 175301-803

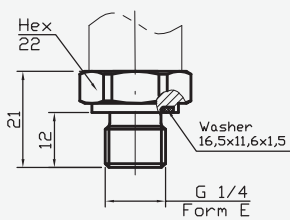


MVS/C  
DIN EN 175301-803

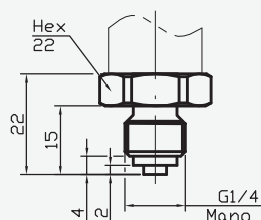


Pressure Connections\*

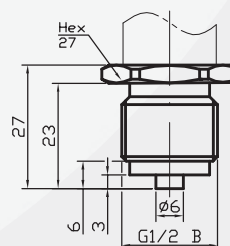
G 1/4 A; DIN 3852; Form E



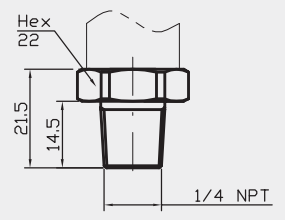
G 1/4 B



G 1/2 B



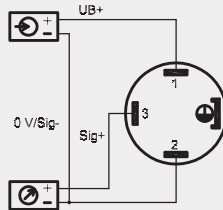
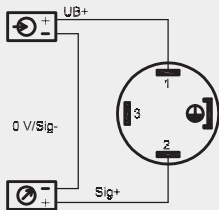
1/4 NPT



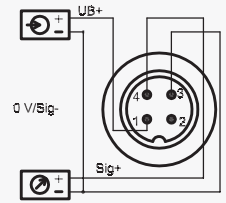
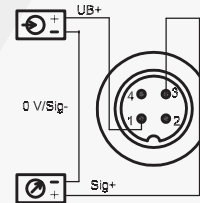
\* custom-made adjustments acc. to pressure connections and connecting options are possible

## Electrical Connections\* (left: 2-wire, right: 3-wire)

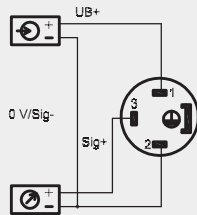
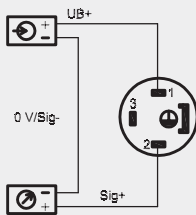
MVS/A  
DIN EN  
175301-803



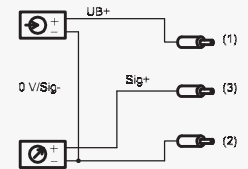
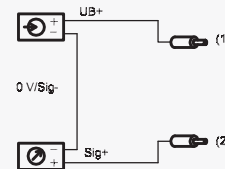
male  
socket  
M12x1  
(S 763)



MVS/C  
DIN EN  
175301-803



cable  
output



### Legend

power supply  
 consumer

(1) red  
(2) black  
(3) white

\* custom-made adjustments acc. to pressure connections and connecting options are possible

### Product line

DS4	Electronic Pressure Switch	SMC	Pressure Transmitter with CANopen Interface
DPSX9I	Intrinsically Safe Electronic Pressure Switch for Current	SME	Pressure Transmitter in Miniature Design
DPSX9U	Intrinsically Safe Electronic Pressure Switch for Voltage	SMF	Pressure Transmitter with Flush Diaphragm
PS1	Level Sensor	SMH	High Pressure Transmitter
PSX2	Intrinsically Safe Level Sensor	SML	Pressure Transmitter for Industrial Application
SHP	High Precision Pressure Transmitter	SMO	Pressure Transmitter in Mobile Hydraulics
SIS	Low Pressure Transmitter in Short and Compact Design	SMS	OEM Pressure Transmitter for Hydraulics and Pneumatics
SIL	Low Pressure Transmitter for Industrial Application	SMX	Intrinsically Safe Pressure Transmitter for Industrial Application
SKE	High Temperature Pressure Transmitter with Detached Electronics	TPS	Multi-Function Transmitter for Pressure and Temperature
SKL	High Temperature Pressure Transmitter with Cooling Fins		



**GUEMISA** (Electrónica Guerra y Miró Guemisa S.L.)  
Sta. Virgilia, 29 - local - 28033 Madrid (Spain)  
Tlfno.: (034) 91 764 21 00 Fax.: (034) 91 764 21 32  
Email.: ventas@guemisa.com Web.: www.guemisa.com

## Información resumida

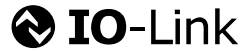
### Medición universal de caudal

- 8 márgenes de medición del caudal de 0,002 l/min hasta 200 l/min
- Gran dinámica de medición (1:50)
- Disponible en unidireccional y bidireccional
- Alta precisión
- Informe de ensayo opcional

### Instalación rápida

- No son necesarios tramos de amortiguación
- Racores QS acodados orientables
- Conectores tipo clavija L1 y M8 para una rápida puesta en funcionamiento

### Interfaz de comunicación



### Práctica forma constructiva

- Diseño compacto de 20x58 mm
- Grado de protección IP 40

### Manejo sencillo

- Pantalla clara con 2 líneas
- Cambio configurable de la pantalla a color rojo
- Guía de menú intuitiva

### Salidas eléctricas conmutables

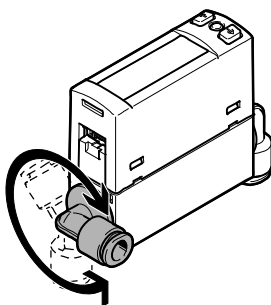
- Diferentes funciones de conmutación
- Salidas de conmutación (PNP/NPN, NO/NC)
- Salidas analógicas (0...10 V, 1...5 V, 4...20 mA)

### Descripción del producto

El sensor de caudal SFAH está diseñado para la supervisión de aire comprimido y gases no corrosivos. Gracias a su diseño compacto, este sensor puede utilizarse en muchos campos. El proceso de medición se basa en el procedimiento térmico Heat Transfer. La construcción en bypass reduce la probabilidad de sufrir anomalías debido a partículas y humedad. El valor del caudal se envía al sistema de mando conectado en forma de señal de conexión, señal analógica o a través de IO-Link.

### Ocupa poco espacio

Racores QS acodados orientables



### Aplicaciones

- Monitorización del proceso
- Manipulación de piezas pequeñas
- Control del consumo de aire comprimido
- Control de estanquidad
- Control del gas protector
- Control neumático de objetos mediante medición aire-ranura

### Funciones

- Supervisión y ajuste de un umbral de caudal, un margen de caudal o una modificación en el caudal
- Control con función de aprendizaje o introduciendo los valores
- Salida de caudal volumétrico y caudal másico en las unidades de flujo habituales
- Función ECO mediante el ajuste de la desconexión de la pantalla
- Código de seguridad opcional y configurable (código de 4 dígitos)
- Filtro paso bajo ajustable para el alisado de la señal de caudal
- Escalado de la salida analógica para aumentar la dinámica de las señales
- Compensación de offset opcional
- Memoria de valores mín./máx.
- Todos los ajustes que se han realizado en un sensor (maestro) se pueden transferir (replicar) a otros sensores con la misma construcción (dispositivos)
- Amplio margen de presión de -0,9 bar hasta 10 bar

### IO-Link

- Comunicación en serie a través de IO-Link® 1.1 integrada
- Transmisión cíclica de dos estados de conmutación y del valor medido del caudal
- Posibilidad de configurar de forma remota el sensor utilizando un maestro IO-Link®
- Cambio sencillo del sensor con parametrización automática
- Es posible la identificación de sensores, la diagnosis y el tech-in a través de IO-Link®

# Sensores de caudal SFAH

Código del producto

SFAH - [ ] - [ ] - [ ] - [ ] - [ ] - [ ] + [ ]

**Código del producto**

SFAH	Sensor de caudal
------	------------------

**Margen de medición del caudal**

0,1 <sup>1)</sup>	Máx. 0,1 l/min
0,5	Máx. 0,5 l/min
1	Máx. 1 l/min
5	Máx. 5 l/min
10	Máx. 10 l/min
50	Máx. 50 l/min
100	Máx. 100 l/min
200	Máx. 200 l/min

**Entrada de caudal**

B	Bidireccional
U	Unidireccional

**Conexión neumática**

Q4	Racor de conexión 4 mm
Q6	Racor de conexión 6 mm
Q8	Racor de conexión 8 mm
G14	G1/4
G18	G1/8

**Tipo de rosca**

	Ninguno
F	Rosca interior

**Sentido de la salida**

S	Recto
AR	Acodado, orientable

**Salida eléctrica 1**

PNLK	PNP, NPN o IO-Link®
------	---------------------

**Salida eléctrica 2**

PNVBA	PNP, NPN 0 ... 10 V, 1 ... 5 V o 4 ... 20 mA
-------	---

**Conexión eléctrica**

L1	Conector tipo clavija con diseño L1
M8	Conector tipo clavija M8

**Certificado**

	Ninguno
T	Informe de ensayo

1) Utilizar rosca interior para requisitos de fugas mínimos en el margen de medición inferior, en combinación con la conexión neumática G1/4 o G1/8.

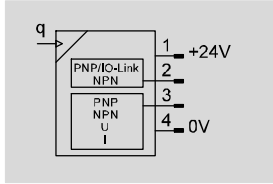
# Sensores de caudal SFAH

Hoja de datos

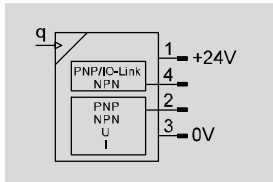
FESTO

Función

SFAH-...-L1



SFAH-...-M8



- Caudal
  - 0,002 ... 0,1 l/min
  - 0,01 ... 0,5 l/min
  - 0,02 ... 1 l/min
  - 0,1 ... 5 l/min
  - 0,2 ... 10 l/min
  - 1 ... 50 l/min
  - 2 ... 100 l/min
  - 4 ... 200 l/min

- Máxima flexibilidad y reducción del número de repuestos gracias a las salidas eléctricas conmutables
- Filtro de señales de medición para ajustar el tiempo de subida
- Filtro adicional para nivelar los valores visualizados



Datos generales especificaciones técnicas	
Homologación	RCM
Marcado CE (ver declaración de conformidad)	Según directiva de máquinas UE CEM Según directiva de máquinas RoHS
Marcado KC	KC-EMV
Nota sobre los materiales	Conformidad con la directiva 2002/95/CE (RoHS)

Señal de entrada, elemento de medición	-0,1	-0,5	-1	-5	-10	-50	-100	-200
Magnitud de medida	Caudal másico, caudal volumétrico							
Sentido de flujo	B	Bidireccional						
	U	Unidireccional						
Principio de medición	Térmico							
Procedimiento de medición	Heat Transfer							
Valor inicial del margen de medición del caudal [l/min]	0,002	0,01	0,02	0,1	0,2	1	2	4
Valor final del margen de medición del caudal <sup>1)</sup> [l/min]	0,1	0,5	1	5	10	50	100	200
Presión de funcionamiento [bar]	-0,9 ... 10							
Medio de funcionamiento	Aire comprimido según ISO 8573-1:2010 [6:4:4]							
	Nitrógeno							
	Argón							
Temperatura del medio [°C]	0 ... 50							
Temperatura ambiente [°C]	0 ... 50							
Temperatura nominal [°C]	23							

1) Para la característica ...B-... :el margen de medición se aplica en sentido positivo y en sentido negativo.

Salida, datos generales	
Precisión del valor de caudal	+/- (2% del valor medio + 1% FS)
Precisión de repetición punto cero [%FS]	±0,2
Margen de la precisión de repetición [% FS]	±0,8
Coefficiente de temperatura margen [% FS/K]	Típicamente ±0,15 (máx. ±0,3)
Caudal margen <sup>1)</sup>	Típicamente ±1 (en el margen de presión -0,7 ... 10 bar)

1) En el margen de presión -0,9 ... -0,7 bar cabe esperar un margen de caudal adicional típicamente de ±4% FS.



# Sensores de caudal SFAH

Hoja de datos

Salida de conmutación	
Salida de conmutación	Conmutable entre 2 x PNP y 2 x NPN
Función de conmutación	Comparador de ventana
	Comparador de valores umbral
	Monitorización automática de la diferencia
Función del elemento de conmutación	Normalmente cerrado/normalmente abierto, conmutable
Corriente máxima de salida [mA]	100

Salida analógica				
Salida analógica	-PNVBA-	0 ... 10 V	1 ... 5 V	4 ... 20 mA
Resistencia de carga máx. en salida de corriente	[ohmios]	500		
Resistencia de carga mín. en salida de tensión	[kiloohmios]	20		

Salida de otros datos	
Resistencia a cortocircuitos	Sí
Resistencia a sobrecarga	Disponible

Dispositivo IO-Link® según IEC 61131-9	
Protocolo	IO-Link®
Versión de protocolo	Dispositivo V 1.1
Perfil	Perfil Smart Sensor
Clases de función	Canal de datos binarios (BDC)
	Variable datos de proceso (PDV)
	Identificación
	Diagnóstico
Teach channel	
Modo de comunicación	COM2 (38,4 kBaud)
Compatibilidad con el modo SIO	Sí
Clase de puerto	A
Ancho de banda de datos de procesos IN	3 bytes
Contenido de datos de proceso IN	2 bits BDC (control de caudal)
	1 bit BDC (monitorización de volumen)
	14 bits PDV (valor medido de caudal)
IO-Link®, contenido de datos de servicio IN	32 bits PDV (valor medido de volumen)
IO-Link®, duración mínima de los ciclos	4 ms
IO-Link®, memoria de datos necesaria	< 0,5 kByte

Electrónica	
Margen de tensión de funcionamiento DC [V]	22 ... 26
Corriente sin carga [mA]	≤ 25
Protección contra inversión de polaridad	en todas las conexiones eléctricas

# Sensores de caudal SFAH

Hoja de datos

<b>Electromecánica</b>	M8	L1
Tipo de conexión	Conectores tipo clavija	
Tecnología de conexión	M8x1, codificación A según EN 61076-2-104	L1J
Cantidad de contactos, hilos	4	

Ocupación de las conexiones		
	Pin	Significado
<b>M8</b>		
	1	Tensión de funcionamiento +24 V
	2	Salida eléctrica 2 (OutB o anlg)
	3	Tensión de funcionamiento 0 V
	4	Salida eléctrica 1 (OutA, cable C/Q para IO-Link®)
<b>L1</b>		
	1	Tensión de funcionamiento +24 V
	2	Salida eléctrica 1 (OutA, cable C/Q para IO-Link®)
	3	Salida eléctrica 2 (OutB o anlg)
	4	Tensión de funcionamiento 0 V

<b>Mecánica</b>	-Q4	-Q6	-Q8	-G18F	-G14F
Tipo de fijación	Con accesorios				
Posición de montaje	Indistinta				
Conexión neumática	Para tubo flexible de diámetro exterior de 4 mm	Para tubo flexible de diámetro exterior de 6 mm	Para tubo flexible de diámetro exterior de 8 mm	Rosca interior G1/8	Rosca interior G1/4
Sentido de la salida	S	Recto			
	AR	Acodado, orientable			-
Peso del producto	[g]	60		90	
<b>Materiales</b>					
Carcasa	Reforzada con poliamida				
Materiales en contacto con el fluido	NBR				
	Silicio				
	Reforzada con poliamida				
	Nitruro de silicio				
	Acero inoxidable de alta aleación				
	Epoxi				
Aleación forjada de aluminio anodizado					

# Sensores de caudal SFAH

Hoja de datos

Indicación/manejo	
Tipo de indicador	LCD iluminado de varios colores
Unidades representables	
0,1	l/min, l/h, scft/h, g/min, l, scft, g
0,5	
1	
5	l/min, l/h, scft/min, scft/h, g/min, l, scft, g
10	
50	l/min, scft/min, scft/h, g/min, l, scft, g
100	
200	
Posibilidades de ajuste	Programación tipo teach-in
	IO-Link®
	Mediante pantalla y teclas
Seguro contra manipulaciones	IO-Link®
	Código PIN

Inmisión/emisión	0,1	0,5	1	5	10	50	100	200
Grado de protección <sup>1)</sup>	IP40							
Fuga máxima permitida [l/h]	0,1							
Caída de presión delta p <sup>2)</sup> [mbar]	< 5					12	15	56
Clase de protección	III							
Clase de resistencia a la corrosión <sup>3)</sup>	2							

1) En combinación con una cubierta en el montaje horizontal, se obtiene el tipo de protección IP54; consulte la ilustración de la página 3.

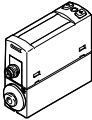
2) Para 6 bar en la entrada y q máx.

3) Clase de resistencia a la corrosión CRC 2 según norma de Festo FN 940070

Componentes con moderado riesgo de corrosión. Aplicación en interiores en caso de condensación. Piezas exteriores visibles con características esencialmente decorativas en la superficie que están en contacto directo con atmósferas habituales en entornos industriales.

# Sensores de caudal SFAH

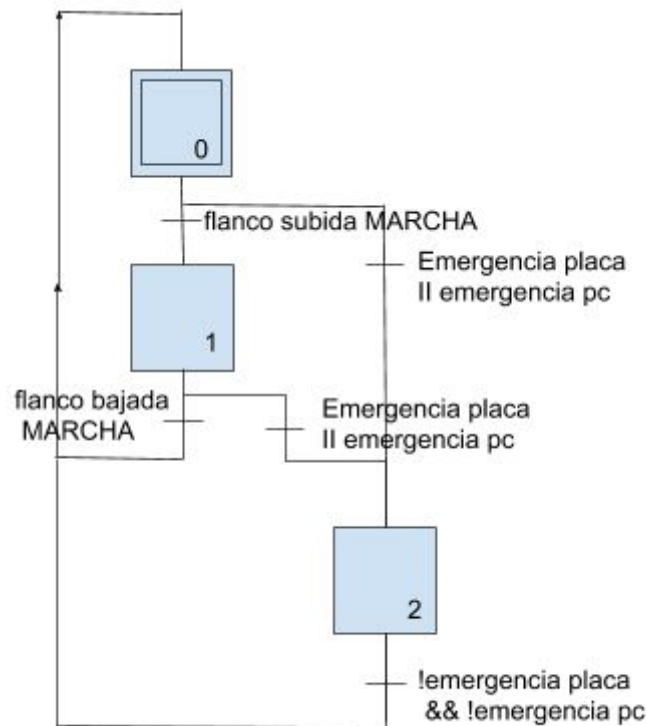
Hoja de datos

Referencias de pedido				
Ejecución	Margen de medición del caudal [l/min]	Tipo de conexión	Nº art.	Código del producto
	0,1 l/min	Para tubo flexible con diámetro exterior de 4 mm	8058461	SFAH-0.1U-Q4S-PNLK-PNVBA-L1
			8058462	SFAH-0.1U-Q4S-PNLK-PNVBA-M8
	0,5 l/min		8058463	SFAH-0.5U-Q4S-PNLK-PNVBA-L1
			8058464	SFAH-0.5U-Q4S-PNLK-PNVBA-M8
	1 l/min		8058465	SFAH-1U-Q4S-PNLK-PNVBA-L1
			8058466	SFAH-1U-Q4S-PNLK-PNVBA-M8
	5 l/min	Para tubo flexible con diámetro exterior de 6 mm	8058467	SFAH-5U-Q6S-PNLK-PNVBA-L1
			8058468	SFAH-5U-Q6S-PNLK-PNVBA-M8
	10 l/min		8058469	SFAH-10U-Q6S-PNLK-PNVBA-L1
			8058470	SFAH-10U-Q6S-PNLK-PNVBA-M8
	50 l/min	Para tubo flexible con diámetro exterior de 8 mm	8058471	SFAH-50U-Q8S-PNLK-PNVBA-L1
			8058472	SFAH-50U-Q8S-PNLK-PNVBA-M8
		Rosca interior G1/8	8058473	SFAH-50U-G18FS-PNLK-PNVBA-M8
	100 l/min	Para tubo flexible con diámetro exterior de 8 mm	8058474	SFAH-100U-Q8S-PNLK-PNVBA-L1
			8058475	SFAH-100U-Q8S-PNLK-PNVBA-M8
		Rosca interior G1/4	8058476	SFAH-100U-G14FS-PNLK-PNVBA-M8
200 l/min	Para tubo flexible con diámetro exterior de 8 mm	8058477	SFAH-200U-Q8S-PNLK-PNVBA-L1	
		8058478	SFAH-200U-Q8S-PNLK-PNVBA-M8	
		8058479	SFAH-200U-G14FS-PNLK-PNVBA-M8	
	Rosca interior G1/4			

## 2.1.4 CÓDIGO ARDUINO

El código arduino que implementa el sistema consta de un esquema grafcet con 3 etapas, una de paro, una de marcha y la última de emergencia. El programa arduino está programado de manera sistemática, constando de 7 puntos indicados y explicados, sin contar la adición de librerías necesarias o la creación de variables.

El esquema al que refiere es el siguiente:



Esquema 6: Esquema Grafcet del sistema

El funcionamiento general del arduino es estando en 0 medir señales pero no actuar sobre las válvulas y los reguladores. Una vez la Marcha se activa, pasamos a la etapa 1 donde el intercambio de información entre el pc y el arduino hace que este controle la actuación de las válvulas y las señales enviadas a los reguladores.

La etapa dos es la etapa de emergencia, que se activa en cualquier caso, activando el estado de emergencia que consta de un cierre específico del sistema y solo pudiendo volver al estado normal de parada cuando la emergencia finaliza.

El código arduino está escrito siguiendo la implementación de automatismos secuenciales en arduino. Sigue el siguiente esquema:

Programa de usuario

1. Inicialización de las etapas (solo una vez).
2. Lectura de entradas

3. Detección de flancos de entradas
4. Desactivación/activación de las etapas
5. Detección de flancos de etapas
6. Activación de salidas
7. Actualización de salidas

Código Arduino:

```
#include <SimpleModbusSlave.h>
```

```
bool E[3] = {1, 0, 0};
```

```
bool flanco_subida_marcha = 0, flanco_bajada_marcha = 0, marcha = 0, copia_marcha = 0, emergencia_pc = 0, emergencia_placa = 0;
```

```
float set_reguladorC_Ox, set_reguladorC_H, set_reguladorP_Ox, set_reguladorP_H, presion_Ox, presion_H, caudal_Ox, caudal_H;
```

```
unsigned long int t_actual, t_purga_Ox, t_purga_H, t_periodo_purga_Ox, t_periodo_purga_H, t1 = 0, t2 = 0, t3 = 0, t4 = 0;
```

```
float ac_valvula_Ox_t, ac_valvula_H_t;
```

```
unsigned long int fin_t_2_05 = 1, fin_t_2_1 = 1, fin_t_2_15 = 1;
```

```
bool t_2_05 = 1, t_2_1 = 1, t_2_15 = 1;
```

```
const int pinH = 2, pinOX = 3, pinHt = 4, pinOXt = 5, pinROC = 6, pinRHC = 7, pinEmer = 8, pinROP = 9, pinRHP = 10;
```

```
float output_set_reg_OX_P, output_set_reg_H_P, output_set_reg_OX_C, output_set_reg_H_C;
```

```
float output_val_OX, output_val_H, output_val_OX_t, output_val_H_t;
```

```
float ac_purga_Ox = 0, ac_purga_H = 0;
```

```
enum
```

```
{
```

```
DIR_VALVULA_OX, //0 -
```

```
DIR_VALVULA_H, //1 -
```

```
DIR_VALVULA_OX_t, //2 -
```

```
DIR_VALVULA_H_t, //3 -
```

```
DIR_PRESION_OX, //4 -
```

```
DIR_PRESION_H, //5 -
```

```
DIR_CAUDAL_OX, //6 -
```

```
DIR_CAUDAL_H, //7 -
```

```
DIR_AC_VALVULA_OX_T, //8
```

```
DIR_AC_VALVULA_H_T, //9
```

```
DIR_SET_REGULADORC_OX, //A -
```

```
DIR_SET_REGULADORC_H, //B -
```

```

    DIR_SET_REGULADORP_OX, //C -
    DIR_SET_REGULADORP_H, //D -
    DIR_EMERGENCIA_PC, //E -
    DIR_T_PURGA_OX, //F -
    DIR_T_PERIODO_PURGA_OX, //10 -
    DIR_T_PURGA_H, //11 -
    DIR_T_PERIODO_PURGA_H, //12 -
    DIR_MARCHA, //13 -
    HOLDING_REGS_SIZE
};
unsigned int holdingRegs[HOLDING_REGS_SIZE];

void setup() {
    modbus_configure(&Serial, 9600, SERIAL_8E1, 1, 13, HOLDING_REGS_SIZE,
holdingRegs);
    modbus_update_comms(9600, SERIAL_8E1, 1);

    pinMode(pinH, OUTPUT); //Válvula Hidrógeno
    pinMode(pinOX, OUTPUT); //Válvula Oxígeno
    pinMode(pinHt, OUTPUT); //Válvula Hidrógeno.T
    pinMode(pinOXt, OUTPUT); //Válvula Oxígeno.T
    pinMode(pinROC, OUTPUT); //Regulador caudal másico Oxígeno
    pinMode(pinRHC, OUTPUT); //Regulador caudal másico Hidrógeno
    pinMode(pinEmer, INPUT_PULLUP); //pulsador emergencia
    pinMode(pinROP, OUTPUT); //Regulador presión Oxígeno
    pinMode(pinRHP, OUTPUT); //Regulador presión Hidrógeno

```

Fuera del esquema, en esta primera parte se crean las variables que se utilizarán en el programa, tanto para los temporizadores, salidas, entradas, flancos. También se incluye la librería SimpleModbusSlave.h, que sirve para la comunicación maestro-esclavo entre el ordenador y la placa. Para ello se crea un vector Enum con X número de directorios que luego serán utilizados por el programa Codesys. Ya por último se configura el modbus para que actualice dicho intercambio de información y se deciden los PIN como salidas o entradas.

### ***// 1. INIALIZACIÓN DE LAS SALIDAS***

```

digitalWrite(pinH, 0);
digitalWrite(pinOX, 0);
digitalWrite(pinHt, 0);
digitalWrite(pinOXt, 0);
digitalWrite(pinROC, 0);

```

```

digitalWrite(pinRHC, 0);
digitalWrite(pinROP, 0);
digitalWrite(pinRHP, 0);

}

```

```

void loop() {

```

En la primera parte del esquema, se inician las salidas a 0 antes de que el programa empiece, por si por algún otro proceso o un apagado forzado estuvieran en una posición no deseada.

```

// 2.LECTURA ENTRADAS

```

```

marcha = holdingRegs[DIR_MARCHA];
emergencia_pc = holdingRegs[DIR_EMERGENCIA_PC];
emergencia_placa = digitalRead(pinEmer);

set_reguladorC_Ox = ((holdingRegs[DIR_SET_REGULADORC_OX]) * 0.818 + 205) /
30;
set_reguladorC_H = ((holdingRegs[DIR_SET_REGULADORC_H]) * 0.818 + 205) / 30;
set_reguladorP_Ox = ((holdingRegs[DIR_SET_REGULADORP_OX]) * 23.72527 + 205)
/ 300;
set_reguladorP_H = ((holdingRegs[DIR_SET_REGULADORP_H]) * 23.72527 + 205) /
300;

//Reinicio de las variables
presion_Ox = 0;
presion_H = 0;
caudal_Ox = 0;
caudal_H = 0;
i=0;

// Filtro mediana, lectura de las presiones y caudales
for (i = 0; i <= 10; i++) {
presion_Ox += analogRead(A3);
presion_H += analogRead(A2);
caudal_Ox += analogRead(A0);
caudal_H += analogRead(A1);
}
presion_Ox = presion_Ox/10;

```



```
presion_H = presion_H/10;  
caudal_Ox = caudal_Ox/10;  
caudal_H = caudal_H/10;
```

```
//caudal 0-100 litros por minuto
```

```
ac_valvula_Ox_t = holdingRegs[DIR_AC_VALVULA_OX_T];  
ac_valvula_H_t = holdingRegs[DIR_AC_VALVULA_H_T];
```

```
t_purga_Ox = holdingRegs[DIR_T_PURGA_OX] * 1000;  
t_periodo_purga_Ox = holdingRegs[DIR_T_PERIODO_PURGA_OX] * 1000;  
t_purga_H = holdingRegs[DIR_T_PURGA_H] * 1000;  
t_periodo_purga_H = holdingRegs[DIR_T_PERIODO_PURGA_H] * 1000;
```

En dicha parte del programa se leen repetidamente los valores de las variables, tanto si llegan del ordenador como si llegan de la placa arduino, es decir de los sensores que llevará conectados. Los caudales y presiones pasan por un filtro de media, que leerá 10 valores y hará la media de ellos antes de enviarlos al ordenador. También se reinician dichos valores para que sea posible el ciclo de lectura permanentemente.

### *// 3.DETECCIÓN DE FLANCOS*

```
flanco_subida_marcha = (!copia_marcha) && marcha;  
flanco_bajada_marcha = (copia_marcha) && !marcha;  
copia_marcha = marcha;
```

En este pequeño sector se escribe lo necesario para crear los flancos de subida o de bajada, que serán utilizados posteriormente.

### *// 4.DESACTIVACIÓN/ACTIVACIÓN ETAPAS*

```
if (E[0] && flanco_subida_marcha) {  
    E[0] = 0;  
    E[1] = 1;  
}  
else if (E[1] && flanco_bajada_marcha) {  
    E[1] = 0;  
    E[0] = 1;  
}  
else if ((E[0]) && (emergencia_pc || emergencia_placa)) {  
    E[0] = 0;
```

```

    E[2] = 1;
    fin_t_2_05 = millis() + 500;
    fin_t_2_1 = millis() + 1000;
    fin_t_2_15 = millis() + 1500;
}
else if ((E[1] && (emergencia_pc || emergencia_placa)) {
    E[1] = 0;
    E[2] = 1;
    fin_t_2_05 = millis() + 500;
    fin_t_2_1 = millis() + 1000;
    fin_t_2_15 = millis() + 1500;
}
else if (E[2] && !emergencia_placa && !emergencia_pc) {
    E[2] = 0;
    E[0] = 1;
}
}

```

En la cuarta parte del programa se escribe el esquema explicado anteriormente, la evolución de las etapas del grafted y se inician los contadores que se utilizarán en dichas etapas.

#### **// 5.DEFINICIÓN TEMPORIZADORES**

```

t_2_05 = !(E[2] && (millis() >= fin_t_2_05));
t_2_1 = !(E[2] && (millis() >= fin_t_2_1));
t_2_15 = !(E[2] && (millis() >= fin_t_2_15));

if ((millis() >= t1) || ac_valvula_Ox_t) {
    t1 = millis() + t_periodo_purga_Ox;
    ac_purga_Ox = 1;
}
else if (millis() >= t1 + t_purga_Ox - t_periodo_purga_Ox) {
    ac_purga_Ox = 0;
}
if ((millis() >= t2) || ac_valvula_H_t) {
    t2 = millis() + t_periodo_purga_H;
    ac_purga_H = 1;
}
else if (millis() >= t2 + t_purga_H - t_periodo_purga_H) {
    ac_purga_H = 0;
}
}

```

En dicha parte se programan los temporizadores que afectarán al proceso.

## *// 6.ACTIVACIÓN DE SALIDAS*

```
output_val_OX = (E[1] || (E[2] * t_2_1));  
output_val_H = (E[1] || (E[2] * t_2_1));  
  
output_val_OX_t = (E[1] * ac_purga_Ox) || (E[2] * t_2_05);  
output_val_H_t = (E[1] * ac_purga_H) || (E[2] * t_2_05);  
  
output_set_reg_OX_P = (E[1] || (E[2] * t_2_15)) * set_reguladorP_Ox;  
output_set_reg_H_P = (E[1] || (E[2] * t_2_15)) * set_reguladorP_H;  
output_set_reg_OX_C = (E[1] || (E[2] * t_2_15)) * set_reguladorC_Ox;  
output_set_reg_H_C = (E[1] || (E[2] * t_2_15)) * set_reguladorC_H;
```

En la penúltima parte del programa se activan las salidas en función de las etapas, variables y/o temporizadores utilizados.

## *// 7.ACTUALIZACIÓN SALIDAS*

```
digitalWrite(pinH, output_val_H);  
digitalWrite(pinOX, output_val_OX);  
digitalWrite(pinHt, output_val_H_t);  
digitalWrite(pinOXt, output_val_OX_t);  
analogWrite(pinROC, output_set_reg_OX_C);  
analogWrite(pinRHC, output_set_reg_H_C);  
analogWrite(pinROP, output_set_reg_OX_P);  
analogWrite(pinRHP, output_set_reg_H_P);  
  
// envío de valores de presión y caudal al ordenador  
if (presion_Ox < 205) {  
    holdingRegs[DIR_PRESION_OX] = 205;  
} else if (presion_Ox >= 0) {  
    holdingRegs[DIR_PRESION_OX] = presion_Ox;  
}  
if (presion_H < 205) {  
    holdingRegs[DIR_PRESION_H] = 205;  
} else if (presion_H >= 0) {  
    holdingRegs[DIR_PRESION_H] = presion_H;  
}  
if (caudal_Ox < 205) {  
    holdingRegs[DIR_CAUDAL_OX] = 205;
```

```

} else if (caudal_Ox >= 0) {
    holdingRegs[DIR_CAUDAL_OX] = caudal_Ox;
}

//envío de situación de las válvulas al pc
holdingRegs[DIR_VALVULA_OX] = output_val_OX;
holdingRegs[DIR_VALVULA_H] = output_val_H;
holdingRegs[DIR_VALVULA_OX_t] = output_val_OX_t;
holdingRegs[DIR_VALVULA_H_t] = output_val_H_t;
}

```

Por último, los valores actualizados de las salidas se envían a su destino, es decir a los PIN que corresponden o al propio PC.

### 2.1.5 CÓDIGO CODESYS

El programa funciona en bloques, que son los siguientes. Pero antes de crear los bloques es necesario configurar el programa para que funcione de la manera deseada. Éstos son los pasos a seguir:

- 1) Crear un Proyecto nuevo: Codesys -> Nuevo proyecto -> proyecto standard .
- 2) Indicar que el proyecto es para ejecutar en PC con Windows (que hará las veces de PLC): Codesys Control Win V3 e indicar que el código del programa principal (PLC\_PRG) será ST.
- 3) En el árbol del proyecto pinchar sobre Device (Codesys Control Win V3) -> agregar dispositivo -> Buses de campo -> Modbus -> Puerto serie Modbus -> Modbus COM Port (el PC será el maestro de las comunicaciones)
- 4). Pinchar sobre Modbus\_COM\_Port, botón derecho -> agregar el dispositivo -> Maestro serie Modbus -> Modbus Master COM Port
- 5) Pinchar sobre Modbus\_Master\_COM\_Port, botón derecho -> agregar dispositivo -> Modbus Esclavo, Puerto COM
- 6) Doble click en Modbus\_COM\_Port del árbol del proyecto para configurarlo. En puerto ponemos el de Arduino. El resto de parámetros se quedan igual, pues son los que se han utilizado en el código de Arduino ( 9600 baudios, paridad EVEN, 8 bits de datos, 1 bit de

parada).

7) Doble click en Modbus\_Maestro\_Puerto\_COM: activamos el reinició de comunicación.

8) Doble click en Modbus\_Esclavo\_Puerto\_COM para configurarlo.

9) Pinchar en la pestaña “configuración de esclavo Modbus”: Dirección de esclavo (ID): 1 (el que hay en el código Arduino).

10) Pinchar en la pestaña “Canal esclavo Modbus”. Pinchar abajo en "Agregar canal"

a. Channel 0 (para lecturas): Read, función Modbus 3, desplazamiento 0 (las lecturas están en las primeras posiciones de Arduino), longitud 2 (hay dos variables a leer).

b. Channel 1 (para escrituras): Write, función Modbus 16, desplazamiento 2 (las escrituras están en la tercera posición de Arduino), longitud 2 (hay dos variables a escribir)

11) Ahora agregar objeto -> POU y crear las funciones necesarias para el desarrollo del programa, que son los siguientes

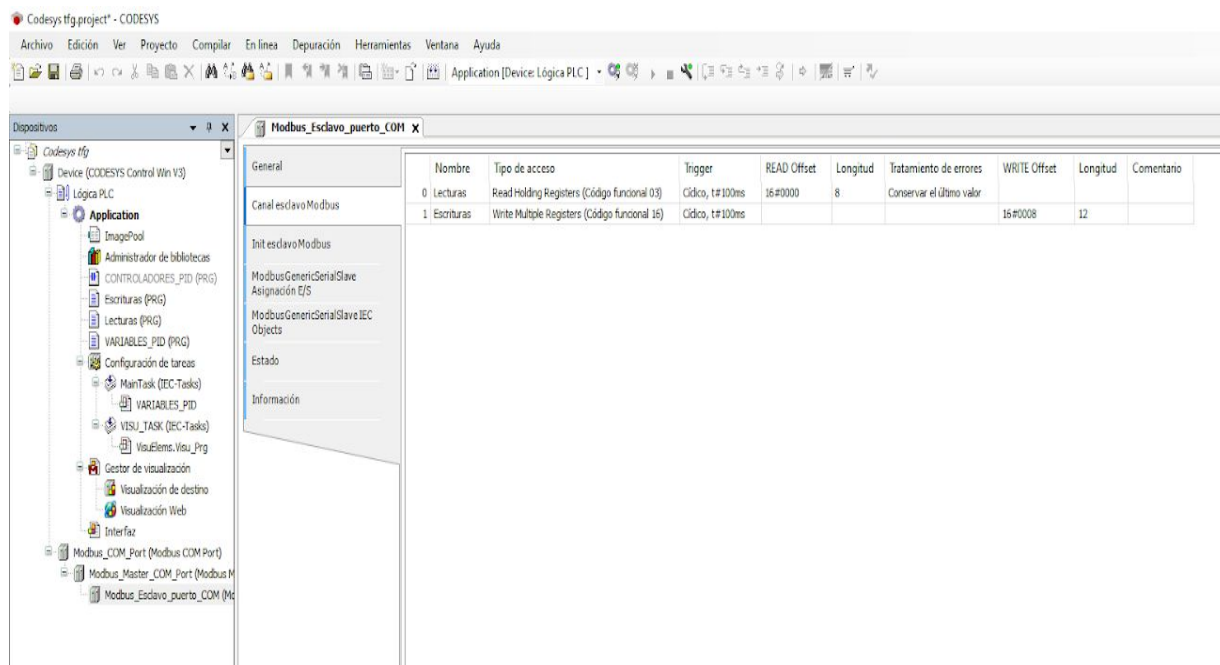


Ilustración 24: Captura de pantalla del programa codesys, sección modbus\_Esclavo\_puerto

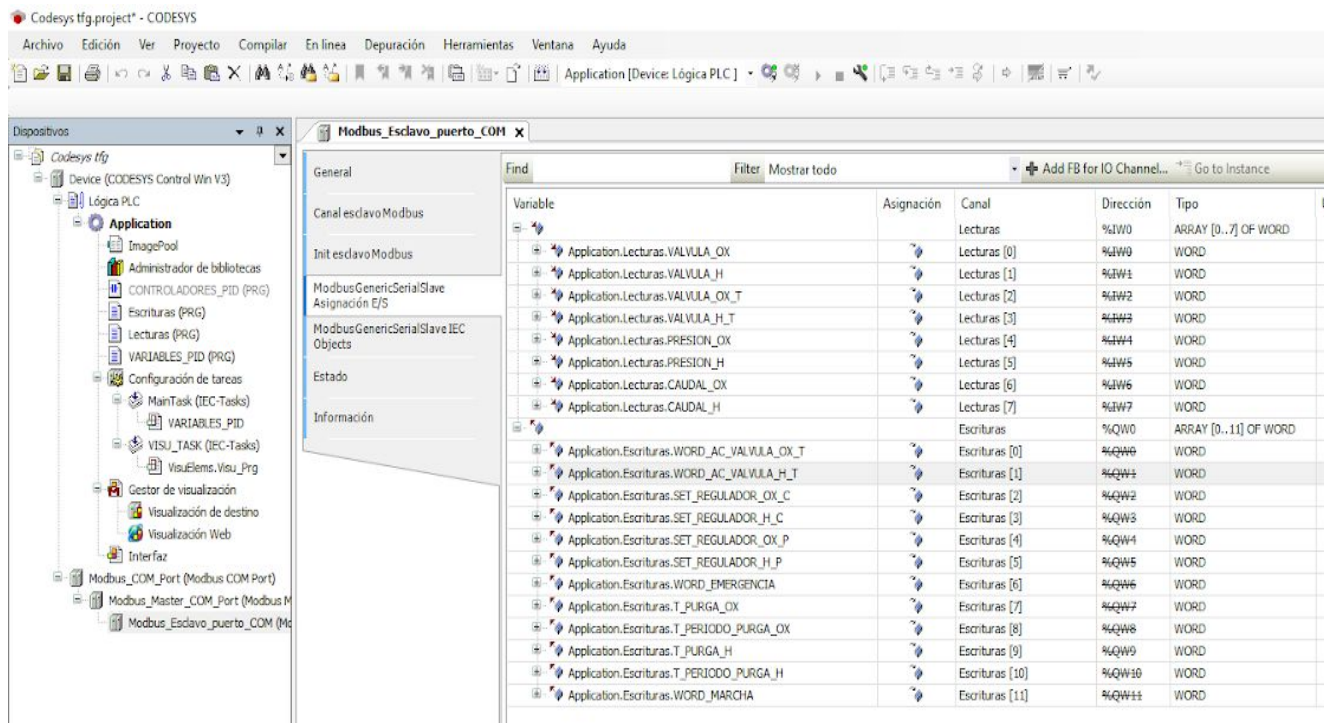


Ilustración 25: Captura de pantalla del programa codesys, sección modbus\_Esclavo\_puerto\_COM

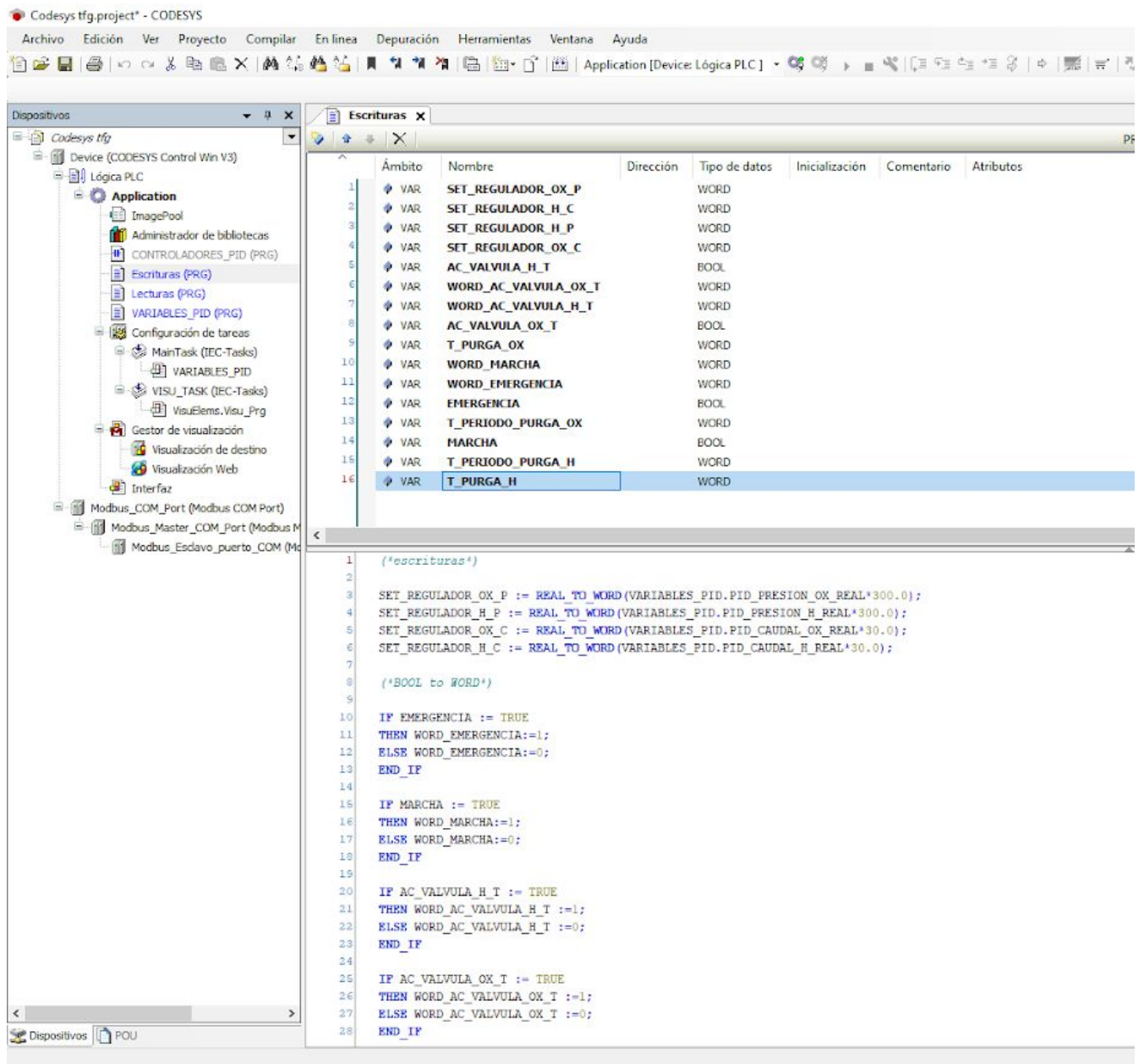


Ilustración 26: Captura de pantalla del programa codesys, sección POU escrituras 1

En el pou de las escrituras han sido creadas las variables con las que el ordenador enviará datos a la placa Arduino. También hay dos cambios de variables, ya que el PID que se verá en los siguientes POU da como salida números REALES, por tanto se convierten en WORD para poder transmitirlos por la placa. A su vez transformamos una variable booleana a word, para poder transmitirla por la placa.



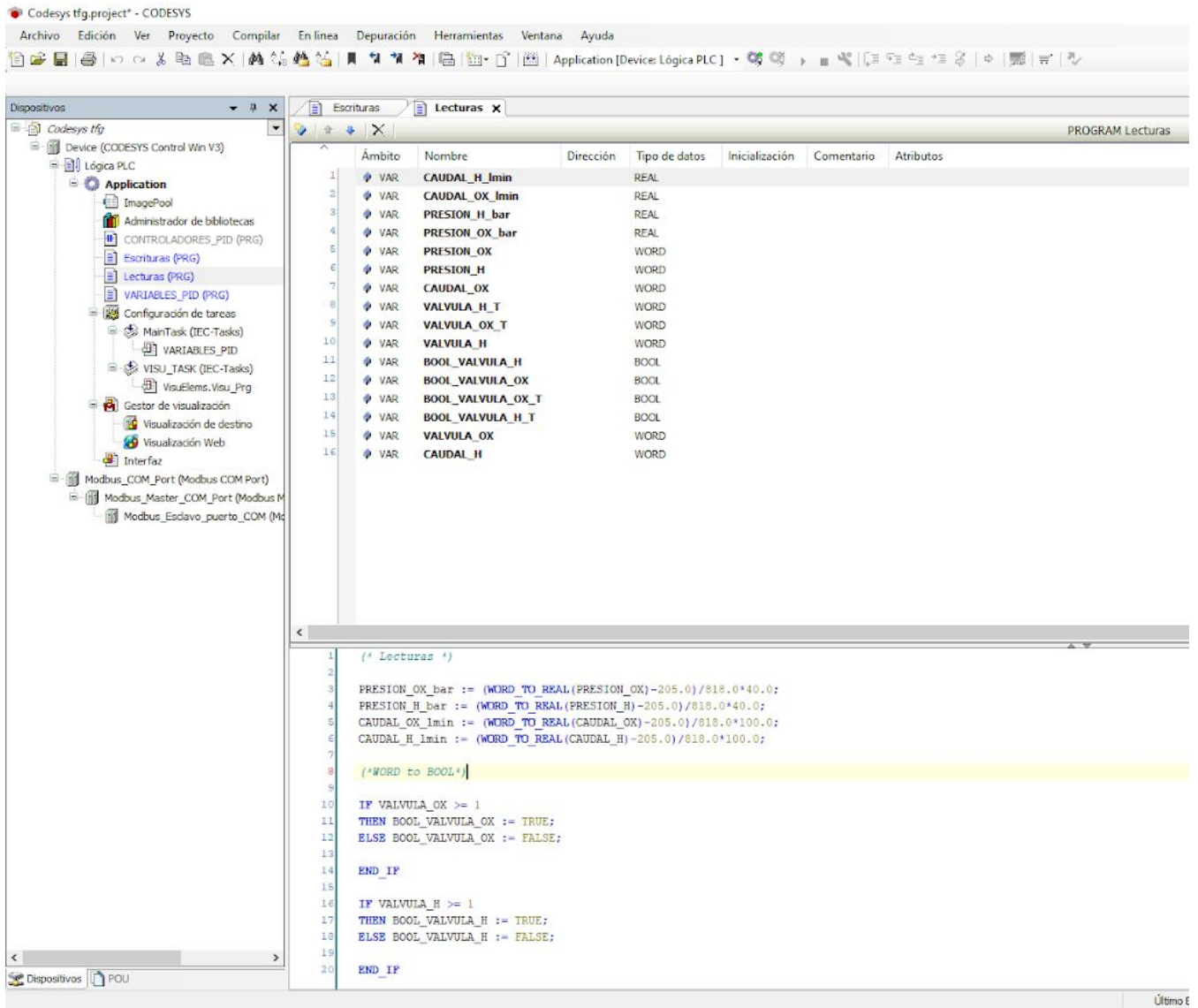


Ilustración 27: Captura de pantalla del programa codesys, sección POU lecturas 1

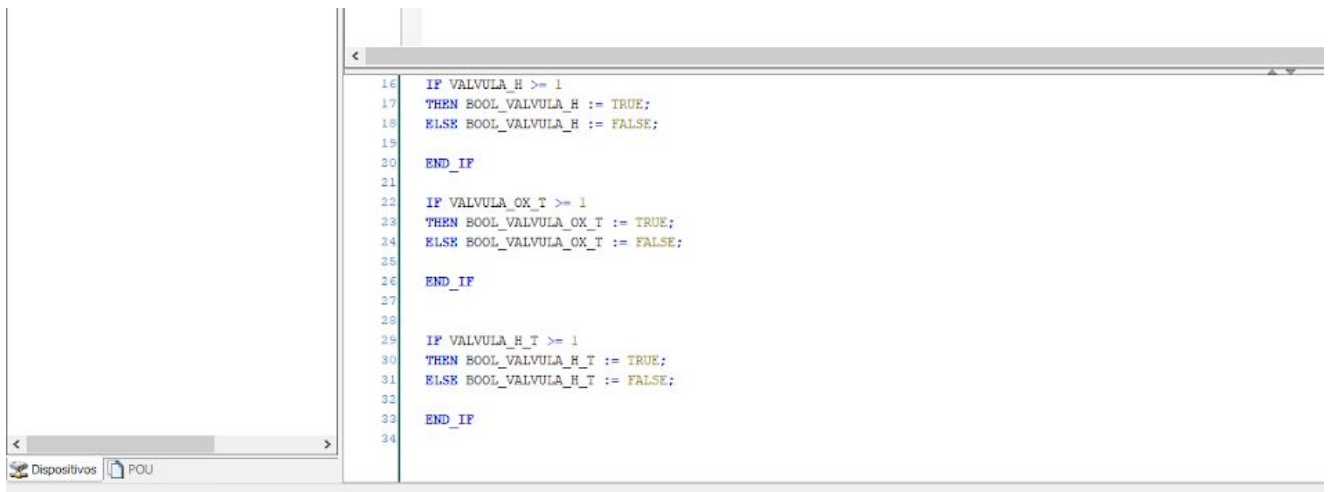


Ilustración 28: Captura de pantalla del programa codesys, sección POU lecturas 2



El POU de las lecturas es muy similar al de las escrituras. Aquí se crean las variables que recibirán datos de la placa arduino, cambiando los valores usados en el PID de WORD a REAL y las lecturas del funcionamiento de las válvulas a BOOL.

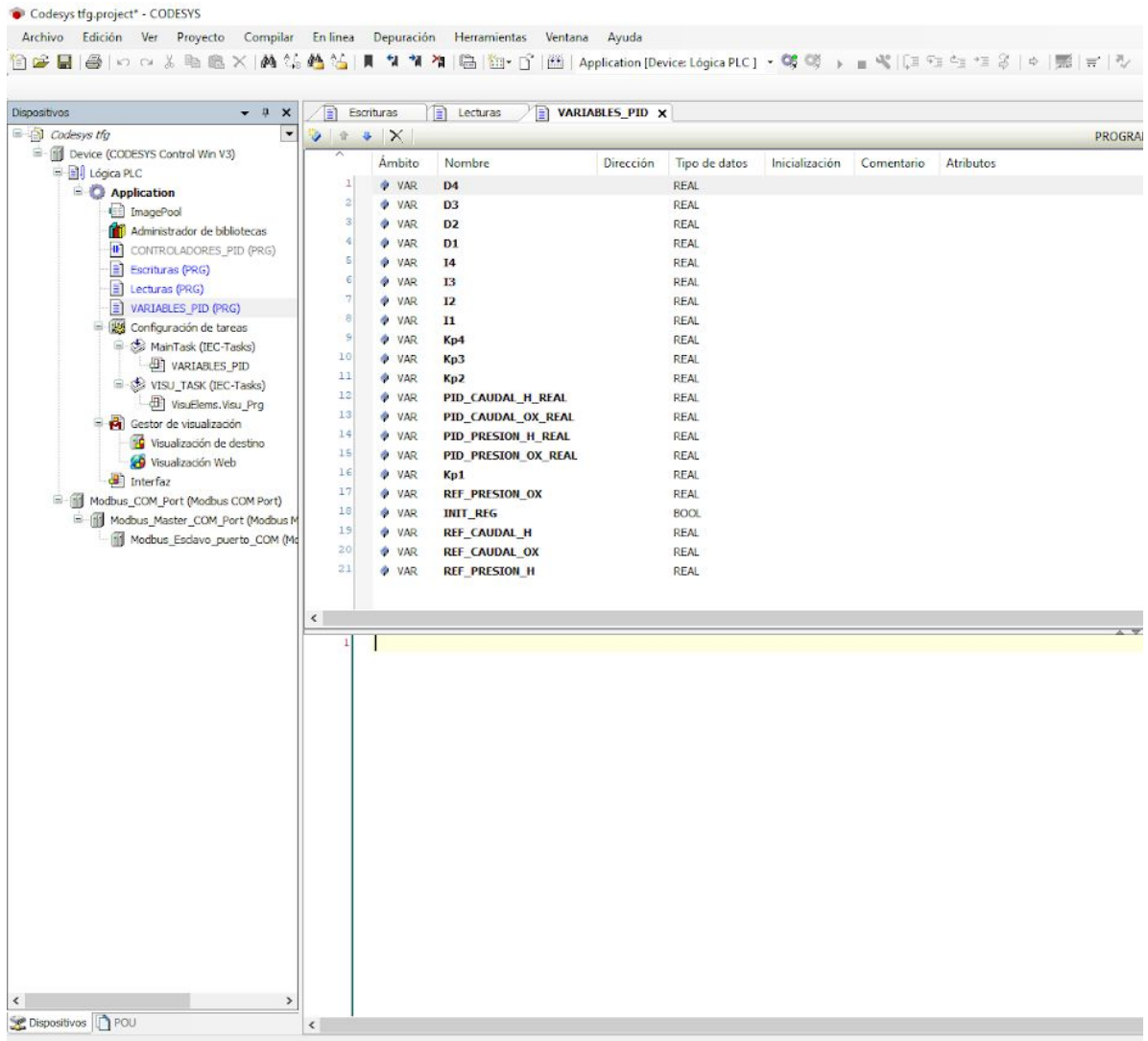


Ilustración 29: Captura de pantalla del programa codesys, sección POU VARIABLES\_PID

Paralelamente se crea un POU para almacenar las variables que serán necesarias en el PID. Contiene la D, Kp y I de cada PID, la referencia a la que deben de llegar y su salida. El ingeniero que trabaje con este proyecto deberá de estudiar los valores más óptimos y darles valores a dichas constantes para el funcionamiento del PID.

Codesys tfg.project - CODESYS

Archivo Edición Ver Proyecto IL/FBD/LD Compilar En línea Depuración Herramientas Ventana Ayuda

Dispositivos Modbus\_Eslavo\_puerto\_COM VARIABLES\_PID CONTROLADORES\_PID x Escrituras

Ámbito	Nombre	Dirección	Tipo de datos	Inicialización	Comentario	Atributos
1	VAR PID_REG_OX_PRESION		Util.PID			
2	VAR PID_REG_H_PRESION		Util.PID			
3	VAR PID_REG_OX_CAUDAL		Util.PID			
4	VAR PID_REG_H_CAUDAL		Util.PID			

Linea	Descripción	Valor	Comentario
1	PID_REG_OX_PRESION		
	Util.PID		
	Lecturas.CAUDAL_OX_lmin	ACTUAL	Y-VARIABLES_PID.PID_PRESION_OX_REAL
	VARIABLES_PID.REF_PRESION_OX	SET_POINT	
	VARIABLES_PID.Kp1	KP	
	VARIABLES_PID.I1	TN	
	VARIABLES_PID.D1	TV	
	0	Y_MANUAL	
	0	Y_OFFSET	
	0	Y_MIN	
	34.4738	Y_MAX	
	Escrituras.MARCHA	MANUAL	
	0	RESET	
2	PID_REG_H_PRESION		
	Util.PID		
	Lecturas.PRESION_H_bar	ACTUAL	Y-VARIABLES_PID.PID_PRESION_H_REAL
	VARIABLES_PID.REF_PRESION_H	SET_POINT	
	VARIABLES_PID.Kp2	KP	
	VARIABLES_PID.I2	TN	
	VARIABLES_PID.D2	TV	
	0	Y_MANUAL	
	0	Y_OFFSET	
	0	Y_MIN	
	34.4738	Y_MAX	
	Escrituras.MARCHA	MANUAL	
	0	RESET	
3	PID_REG_OX_CAUDAL		
	Util.PID		
	Lecturas.CAUDAL_OX_lmin	ACTUAL	Y-VARIABLES_PID.PID_CAUDAL_OX_REAL

Ilustración 30: Captura de pantalla del programa codesys, sección POU CONTROLADORES\_PID 1

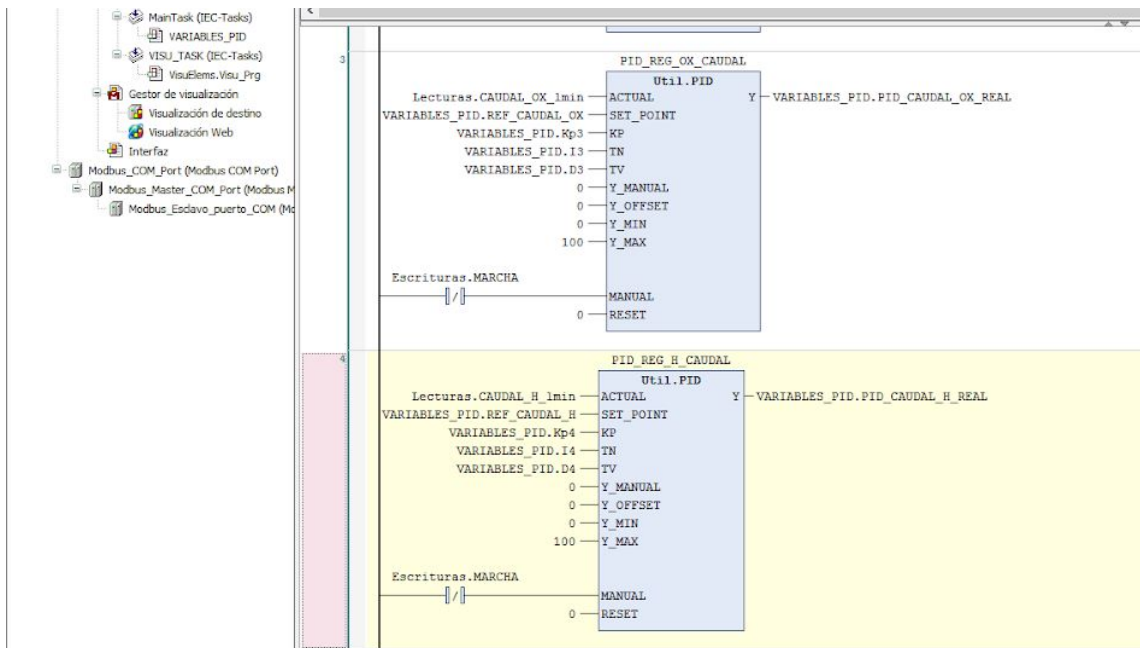


Ilustración 31: Captura de pantalla del programa codesys, sección POU CONTROLADORES\_PID 2

El POU de controladores\_PID contiene los controladores únicamente, y las variables utilizadas se usan de los otros POU creados. Cada PID regula una cosa y solo de oxígeno o hidrógeno, es decir hay dos PID para tanto el caudal como la presión de oxígeno como hay dos PID para controlar tanto la presión como el caudal de hidrógeno.

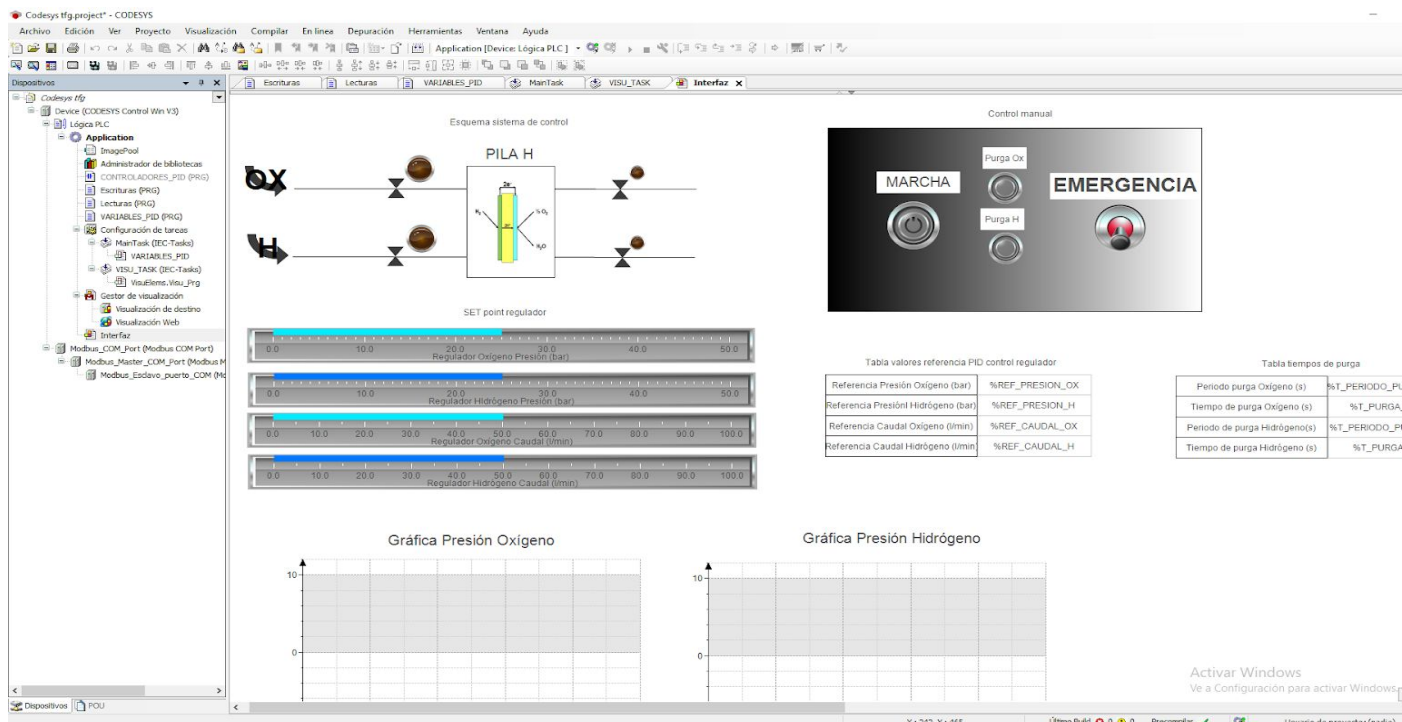


Ilustración 32: Captura de pantalla del programa codesys, sección interfaz 1

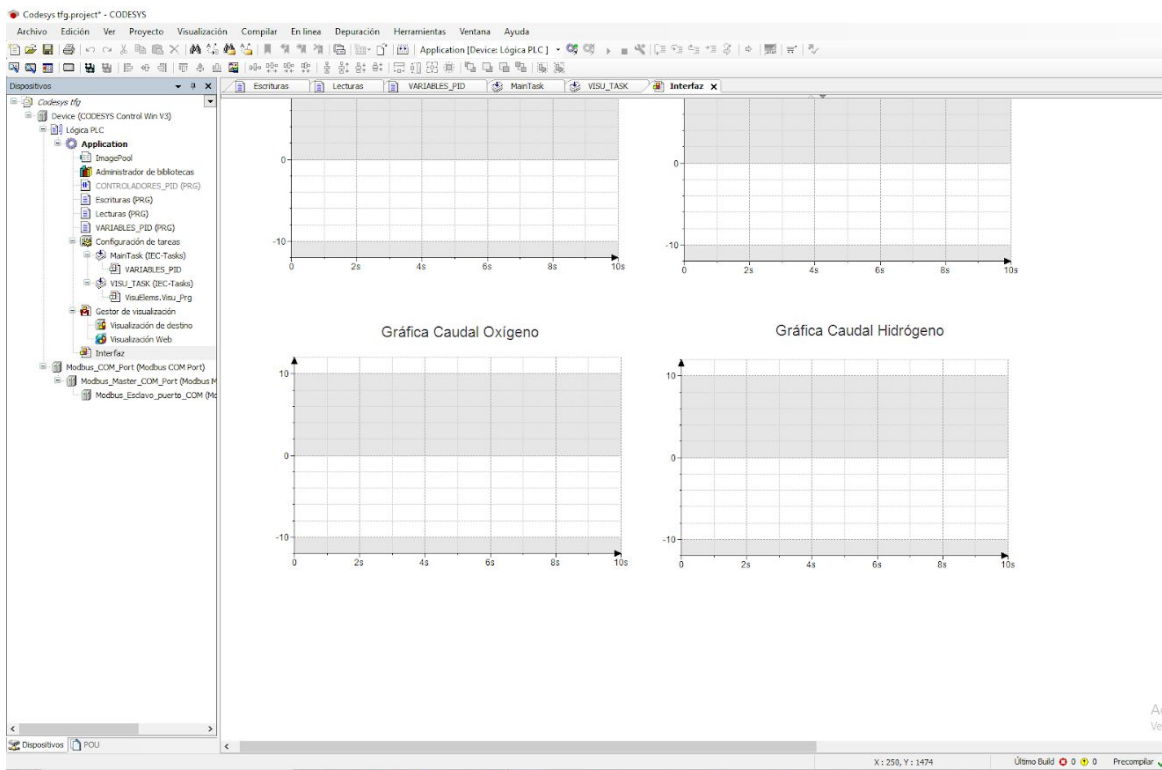


Ilustración 33: Captura de pantalla del programa codesys, sección interfaz 2

Por último el POU de la interfaz, que consta de un dibujo orientativo con 4 leds de iluminación que representan el funcionamiento de las válvulas. A su vez a la derecha una botonera con los botones de marcha, emergencia y purga manual, diferenciando la purga entre oxígeno e hidrógeno. Debajo del esquema hay unas barras que indican el valor que está tomando el regulador actualmente y a su derecha una serie de tablas que debe de rellenar el trabajador con la referencia deseada tanto de presión o caudal de oxígeno y otra tabla con los tiempos para realizar las purgas.

Como parte más grande de la interfaz encontramos las 4 tablas de presión, caudal tanto de oxígeno como de hidrógeno, que contienen las 3 variables de cada una más importantes: la referencia, el valor actual y el valor de salida. Todo esto en tiempo real y con la posibilidad de recogida de datos para su posterior estudio.

VARIABLES UTILIZADAS EN EL PROGRAMA CODESYS:

*Escrituras:*

**SET\_REGULADOR\_OX\_P:** es la variable que contiene el número WORD de la presión del oxígeno, siendo una referencia que se enviará por la placa para que el regulador llegue a dicho valor.

**SET\_REGULADOR\_OX\_C:** es la variable que contiene el número WORD del caudal del oxígeno, siendo una referencia que se enviará por la placa para que el regulador llegue a dicho valor.

**SET\_REGULADOR\_H\_P:** es la variable que contiene el número WORD de la presión del hidrógeno, siendo una referencia que se enviará por la placa para que el regulador llegue a dicho valor.

**SET\_REGULADOR\_H\_C:** es la variable que contiene el número WORD de caudal del hidrógeno, siendo una referencia que se enviará por la placa para que el regulador llegue a dicho valor.

**AC\_VALVULA\_OX\_T:** número BOOLEANO que refiere al botón de la interfaz que controla la actuación sobre la válvula temporizada de oxígeno manualmente.

**AC\_VALVULA\_H\_T:** número BOOLEANO que refiere al botón de la interfaz que controla la actuación sobre la válvula temporizada de hidrógeno manualmente.

**WORD\_AC\_VALVULA\_OX\_T:** variable que recoge la transformación de BOOL a WORD de AC\_VALVULA\_OX\_T y poder transmitirla a la placa.

**WORD\_AC\_VALVULA\_H\_T:** variable que recoge la transformación de BOOL a WORD de AC\_VALVULA\_H\_T y poder transmitirla a la placa.

**T\_PURGA\_OX:** tiempo que dura la purga del oxígeno en WORD, recogido en la tabla de la interfaz y enviado a la placa a través de dicha variable.

**T\_PURGA\_H:** tiempo que dura la purga del hidrógeno en WORD, recogido en la tabla de la interfaz y enviado a la placa a través de dicha variable.

**T\_PERIODO\_PURGA\_OX:** tiempo que transcurre entre purgas de oxígeno. Esta variable almacena en WORD el número recogido en la tabla de la interfaz y lo transmite al arduino.

**T\_PERIODO\_PURGA\_H:** tiempo que transcurre entre purgas de hidrógeno. Esta variable almacena en WORD el número recogido en la tabla de la interfaz y lo transmite al arduino.

**MARCHA:** variable booleana que hace referencia al botón de inicio de la botonera.

**EMERGENCIA:** variable booleana que hace referencia a la palanca de emergencia de la botonera.

**WORD\_MARCHA:** variable que recoge en WORD el valor de MARCHA y poder transmitirlo a la placa.

**WORD\_EMERGENCIA:** variable que recoge en WORD el valor de EMERGENCIA y poder transmitirlo a la placa.

### *Lecturas:*

**PRESION\_OX:** recoge el valor de la presión del oxígeno, enviado desde la placa en 0-1024 bits.

**CAUDAL\_OX:** recoge el valor del caudal del oxígeno, enviado desde la placa en 0-1024 bits.

**PRESION\_H:** recoge el valor de la presión del hidrógeno, enviado desde la placa en 0-1024 bits.

**CAUDAL\_H:** recoge el valor del caudal del hidrógeno, enviado desde la placa en 0-1024 bits.

**PRESION\_OX\_bar:** transforma el variable de PRESION\_OX que es WORD en REAL y con un rango de 0-40 bar.

**PRESION\_H\_bar:** transforma el variable de PRESION\_H que es WORD en REAL y con un rango de 0-40 bar.

**CAUDAL\_OX\_l/min:** transforma el variable de CAUDAL\_OX que es WORD en REAL y con un rango de 0-100 litros/min.

**CAUDAL\_H\_l/min:** transforma el variable de CAUDAL\_H que es WORD en REAL y con un rango de 0-100 litros/min.

**VALVULA\_OX:** recoge la información de si está encendida o apagada la válvula del oxígeno, una información transmitida desde la placa, se almacena en WORD.

**VALVULA\_H:** recoge la información de si está encendida o apagada la válvula del hidrógeno, una información transmitida desde la placa, se almacena en WORD.

**VALVULA\_OX\_T:** recoge la información de si está encendida o apagada la válvula del oxígeno temporizada, una información transmitida desde la placa, se almacena en WORD.

**VALVULA\_H\_T:** recoge la información de si está encendida o apagada la válvula del hidrógeno temporizada, una información transmitida desde la placa, se almacena en WORD.

**BOOL\_VALVULA\_OX:** contiene la transformación de WORD to BOOL de VALVULVA\_OX y dicho valor será utilizado en los LEDs de la interfaz, señalización del funcionamiento de las válvulas.

**BOOL\_VALVULA\_H:** contiene la transformación de WORD to BOOL de VALVULVA\_H y dicho valor será utilizado en los LEDs de la interfaz, señalización del funcionamiento de las válvulas.

**BOOL\_VALVULA\_OX\_T:** contiene la transformación de WORD to BOOL de VALVULVA\_OX\_T y dicho valor será utilizado en los LEDs de la interfaz, señalización del funcionamiento de las válvulas.

**BOOL\_VALVULA\_H\_T:** contiene la transformación de WORD to BOOL de VALVULVA\_H\_T y dicho valor será utilizado en los LEDs de la interfaz, señalización del funcionamiento de las válvulas.

### *Variables PID:*

**Kp1:** constante de proporcionalidad del PID1, del regulador de presión del oxígeno.



**Kp2**: constante de proporcionalidad del PID2, del regulador de presión del hidrógeno.  
**Kp3**: constante de proporcionalidad del PID3, del regulador del caudal del oxígeno.  
**Kp4**: constante de proporcionalidad del PID4, del regulador de caudal del hidrógeno.  
**D1**: hace referencia a TV del PID1, siendo el tiempo de acción derivada, de la parte D en segundos.  
**D2**: hace referencia a TV del PID2, siendo el tiempo de acción derivada, de la parte D en segundos.  
**D3**: hace referencia a TV del PID3, siendo el tiempo de acción derivada, de la parte D en segundos.  
**D4**: hace referencia a TV del PID4, siendo el tiempo de acción derivada, de la parte D en segundos.  
**I1**: hace referencia a TV del PID1, siendo el tiempo de ganancia recíproca, de la parte I en segundos.  
**I2**: hace referencia a TV del PID1, siendo el tiempo de ganancia recíproca, de la parte I en segundos.  
**I3**: hace referencia a TV del PID1, siendo el tiempo de ganancia recíproca, de la parte I en segundos.  
**I4**: hace referencia a TV del PID1, siendo el tiempo de ganancia recíproca, de la parte I en segundos.  
**PID\_PRESION\_OX\_REAL**: es el valor de salida de presión de oxígeno, calculado por el PID. El valor está en REAL.  
**PID\_PRESION\_H\_REAL**: es el valor de salida de presión de hidrógeno, calculado por el PID. El valor está en REAL.  
**PID\_CAUDAL\_OX\_REAL**: es el valor de salida de caudal de oxígeno, calculado por el PID. El valor está en REAL.  
**PID\_CAUDAL\_H\_REAL**: es el valor de salida de caudal de hidrógeno, calculado por el PID. El valor está en REAL.

#### ***CONTROLADORES\_PID:***

**PID\_REG\_OX\_PRESION**: módulo PID usado para regular la presión del oxígeno.  
**PID\_REG\_OX\_CAUDAL**: módulo PID usado para regular la presión del oxígeno.  
**PID\_REG\_H\_PRESION**: módulo PID usado para regular la presión del oxígeno  
**PID\_REG\_H\_CAUDAL**: módulo PID usado para regular la presión del oxígeno.

## 2.1.6 SHIELD ARDUINO

El shield diseñado para la placa Arduino Due encaja en los PIN PWM de la placa y los PIN de entrada analógicos, sumados a los de potencia GND, 5V, 3.3V y RESET. Ignora la parte de PIN digitales.

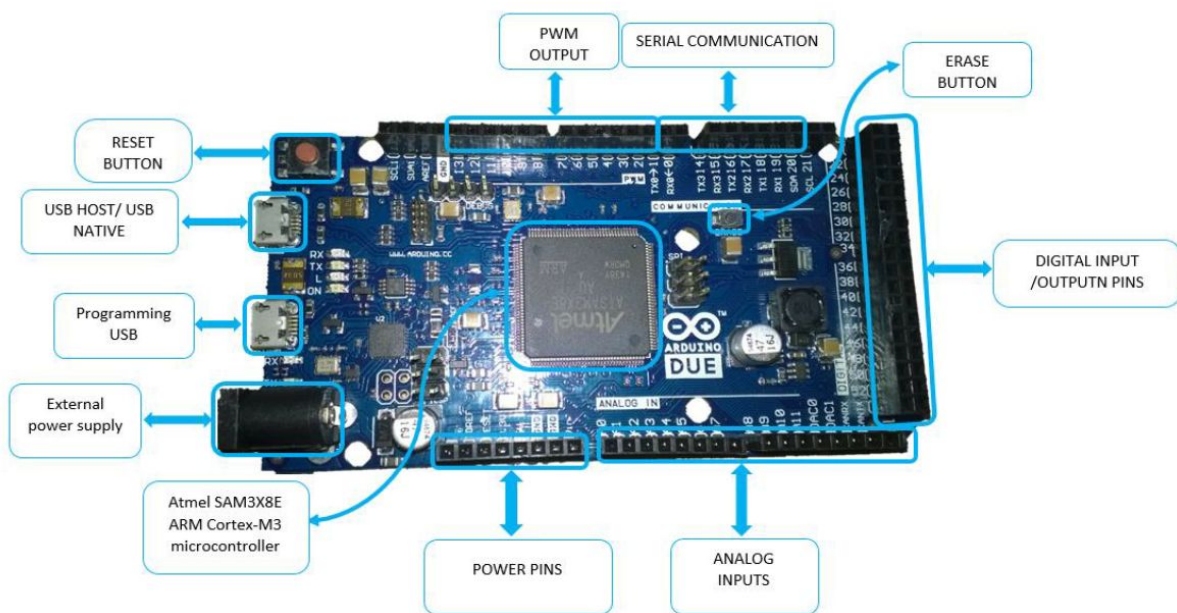


Ilustración 34: Composición placa Arduino DUE

El listado de PIN utilizados y su función es la siguiente:

**PIN\_2\_PWM:** segundo PIN de la parte PWM que hace referencia a la válvula que controla el hidrógeno. Actúa sobre el BJT que regula el contactor de 24V.

**PIN\_3\_PWM:** tercer PIN de la parte PWM que hace referencia a la válvula que controla el oxígeno. Actúa sobre el BJT que regula el contactor de 24V.

**PIN\_4\_PWM:** cuarto PIN de la parte PWM que hace referencia a la válvula que controla las purgas del hidrógeno. Actúa sobre el BJT que regula el contactor de 24V.

**PIN\_5\_PWM:** quinto PIN de la parte PWM que hace referencia a la válvula que controla las purgas del oxígeno. Actúa sobre el BJT que regula el contactor de 24V.

**PIN\_6\_PWM:** sexto PIN de la parte PWM que hace referencia al controlador del regulador de caudal 4-20mA del hidrógeno.

**PIN\_7\_PWM:** séptimo PIN de la parte PWM que hace referencia al controlador del regulador de caudal 4-20 mA del oxígeno.



**PIN\_8\_PWM:** entrada digital del pulsador de emergencia. Aquí se debe de conectar el pulsador, ya que no tiene bornera como el resto de PIN que van directamente a ellas.

**PIN\_9\_PWM:** noveno PIN de la parte PWM que hace referencia al controlador del regulador de presión 4-20 mA del hidrógeno.

**PIN\_10\_PWM:** décimo PIN de la parte PWM que hace referencia al controlador del regulador de presión 4-20 mA del oxígeno.

**PIN\_0\_AN:** PIN número 0 analógico que mide el valor en 4-20 mA o 0-5V del caudal de hidrógeno.

**PIN\_1\_AN:** PIN número 1 analógico que mide el valor en 4-20 mA o 0-5V del caudal de oxígeno.

**PIN\_2\_AN:** PIN número 2 analógico que mide el valor en 4-20 mA o 0-5V de la presión de hidrógeno.

**PIN\_3\_AN:** PIN número 3 analógico que mide el valor en 4-20 mA o 0-5V de la presión de oxígeno.

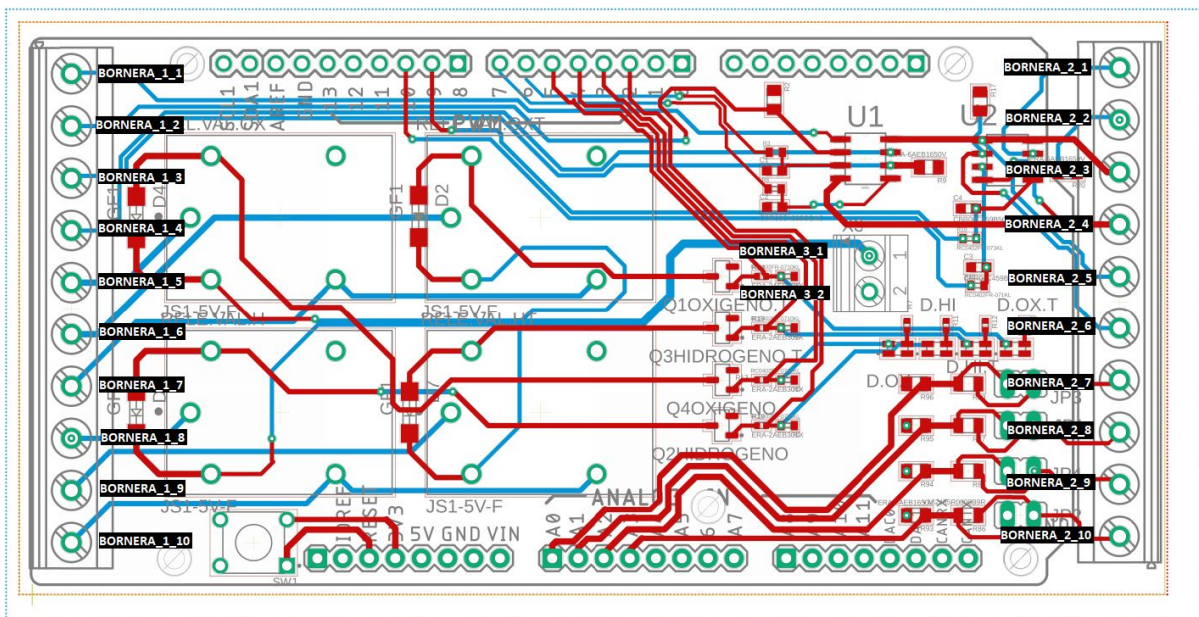


Ilustración 35: Diseño PCB realizado en el TFG con indicación de las borneras

**BORNERA\_1\_1:** Conexión negativa del regulador del caudal de oxígeno.

**BORNERA\_1\_2:** Conexión positiva del regulador del caudal de oxígeno.

**BORNERA\_1\_3:** Conexión negativa del regulador del caudal de hidrógeno.

**BORNERA\_1\_4:** Conexión positiva del regulador del caudal de hidrógeno.

**BORNERA\_1\_5:** Conexión positiva válvula de oxígeno, controlado por el contactor. Funciona a 24V.

**BORNERA\_1\_6:** Conexión positiva válvula temporizada para purga de oxígeno, controlado por el contactor. Funciona a 24V.

**BORNERA\_1\_7:** Conexión negativa de las válvulas de oxígeno.

**BORNERA\_1\_8:** Conexión positiva válvula de hidrógeno, controlado por el contactor. Funciona a 24V.

**BORNERA\_1\_9:** Conexión positiva válvula temporizada para purgas de hidrógeno controlado por el contactor. Funciona a 24V.

**BORNERA\_1\_10:** Conexión negativa de las válvulas de hidrógeno.

**BORNERA\_2\_1:** Bornera de conexión sensor presión de oxígeno.

**BORNERA\_2\_2:** Bornera de conexión sensor presión de hidrógeno.

**BORNERA\_2\_3:** Bornera de conexión sensor caudal de oxígeno.

**BORNERA\_2\_4:** Bornera de conexión sensor caudal de hidrógeno.

**BORNERA\_2\_5:** Conexión positiva del regulador de la presión del oxígeno.

**BORNERA\_2\_6:** Conexión negativa del regulador de la presión del oxígeno.

**BORNERA\_2\_7:** Alimentación positiva amplificador operacional.

**BORNERA\_2\_8:** Alimentación negativa amplificador operacional.

**BORNERA\_2\_9:** Conexión positiva del regulador de la presión del hidrógeno.

**BORNERA\_2\_10:** Conexión positiva del regulador de la presión del hidrógeno.

**BORNERA\_3\_1:** Alimentación positiva transistores BJT.

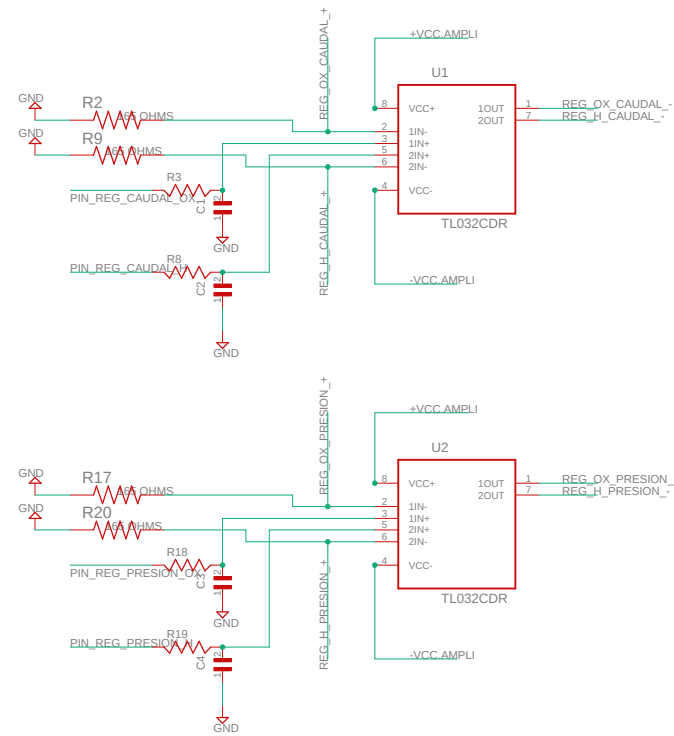
**BORNERA\_3\_2:** Alimentación negativa transistores BJT.



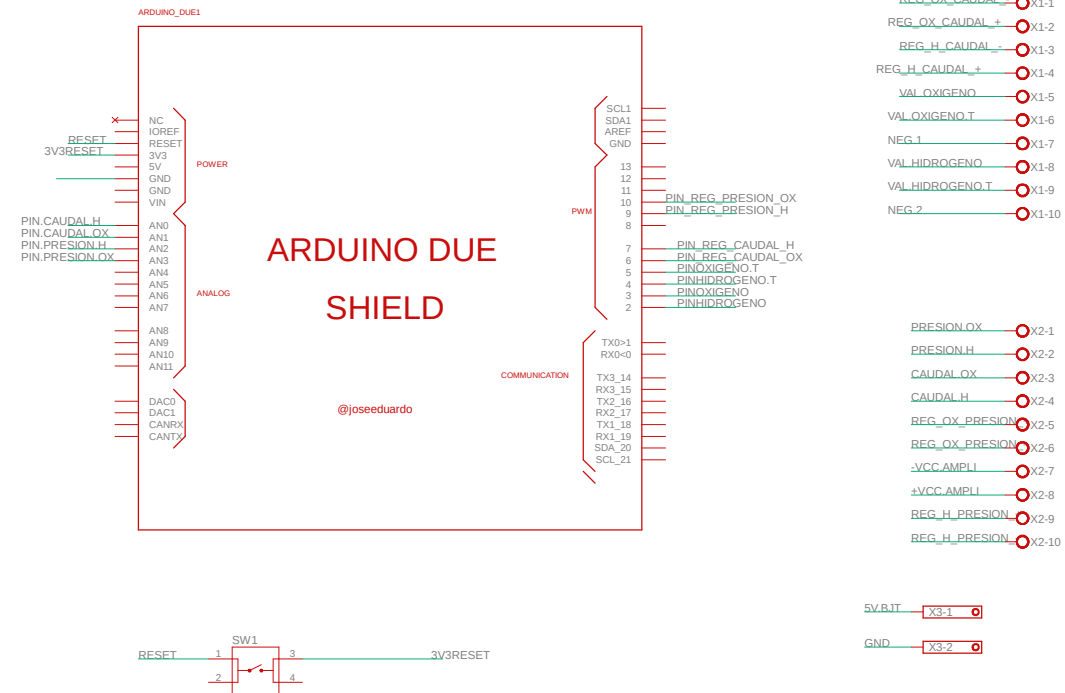
# **PLANOS**



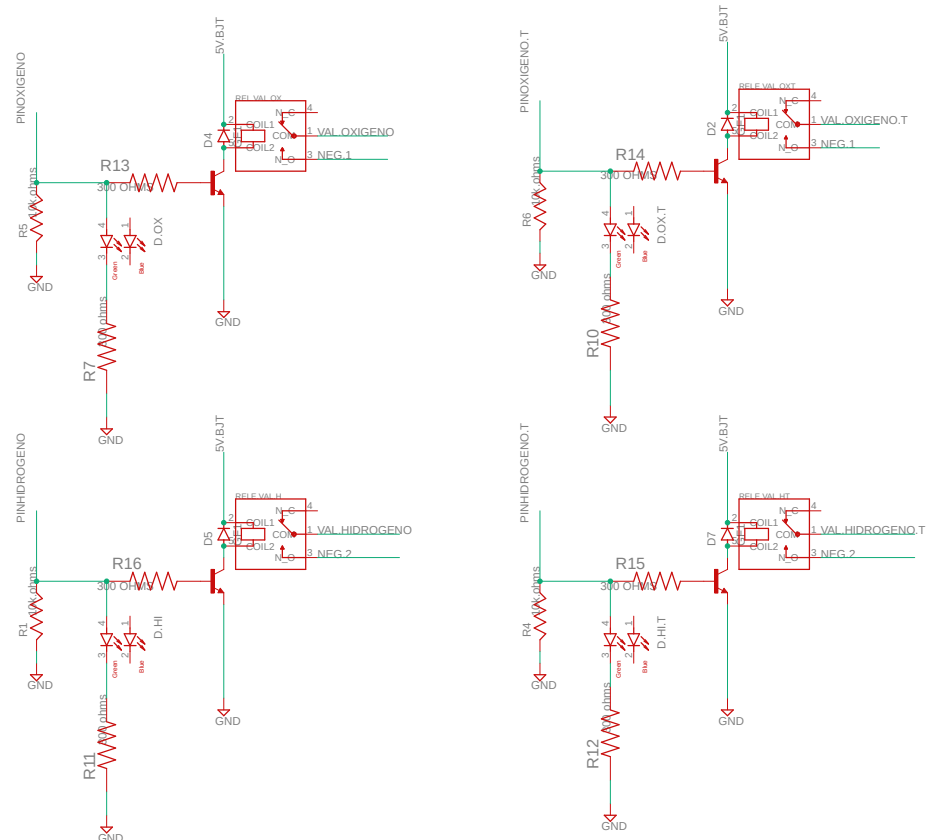
### Salida regulador másico O, H



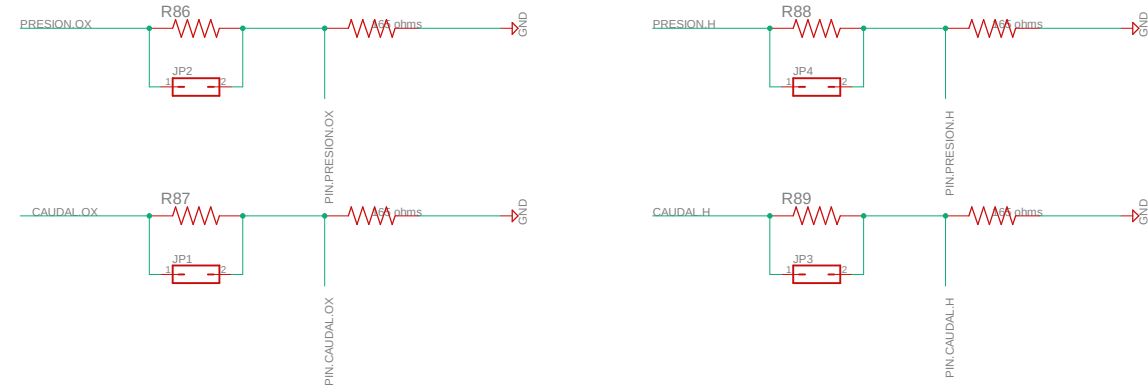
### Arduino Shield Esquema



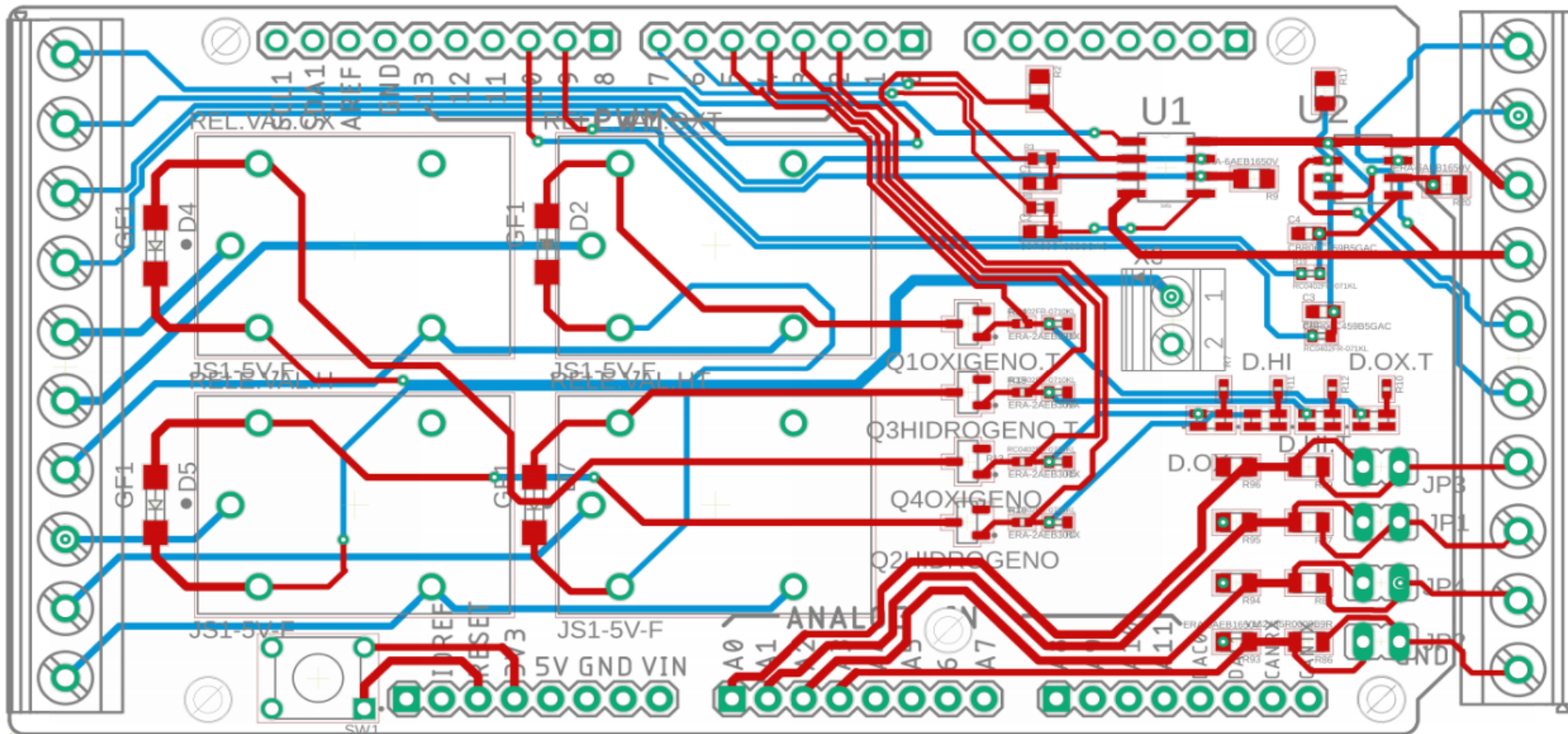
### Control Válvulas O,H, OT, HT.



### Lectura caudalímetro y presión O, H



TITLE: Diadrama Esquemático PCB Shield Arduino DUE	
Document Number:	REV:
Date: 20/08/2020 20:10	Sheet: 1/1



TFG Adrián Carrión Tellols	
TITLE: Placa PCB Shield Arduino Due	
Document Number:	REV:
Date: 20/08/2020 20:10	Sheet: 1/1





# **PLIEGO DE** **CONDICIONES**



## 4.1 REQUISITOS NECESARIOS PARA LA REALIZACIÓN DEL PROYECTO

### Hardware

- PC con monitor, teclado y ratón o touchpad.
- Fuente/s de alimentación variable para los transistores 5V DC, electroválvulas 24 DC y amplificadores +/- 12V DC.
- Laboratorio acondicionado para trabajar con hidrógeno (gas explosivo) y oxígeno (gas altamente inflamable).
- Estructura para el acondicionamiento de todos los elementos del proyecto sumado a la pila de combustible.

### Software

- Windows 7
- Arduino IDE 1.8.12 o en su lugar la última versión disponible.
- Codesys V3.5 SP15 Patch 4 o en su lugar la última versión disponible.

### Adquisición de datos:

Para este proyecto es necesaria la adquisición de los derechos o el diseño de un sistema automático de adquisición de datos de la pila de combustible, siendo la intensidad, el voltaje y la temperatura de ella. Es muy importante para un estudio completo y la posibilidad de mejora de las pilas de combustible de hidrógeno.

## 4.2 MANUAL DE UTILIZACIÓN

- 1° Instalar correctamente las versiones descritas en el proyecto de los programas arduino y codesys.
- 2° Conectar como se ha descrito en el proyecto los componentes a la placa PCB, que va enganchada justo encima de la placa Arduino Due.
- 3° Comprobar que no hay fugas en ninguno de los componentes.
- 4° Compilar el programa arduino y enviarlo a la placa.
- 5° Compilar el programa Codesys.
- 6° En la botonera inferior derecha de Windows, darle a Start a Codesys Control.
- 7° Darle a Run en el programa codesys.
- 8° Utilizar la interfaz como se ha descrito en la sección de resultados finales, pudiendo controlar el sistema y extraer datos.



# **MEDICIONES**



<b>Componentes sensores y actuadores</b>	
<b>Nombre</b>	<b>Cantidad</b>
<b>Regulador 840 &amp; 830 Side-Track® Analog Mass Flow Controller/Meter</b>	2
<b>Placa Arduino Due</b>	1
<b>SIL sensor presión 40 bar</b>	2
<b>Sensor de caudal SFAH</b>	2
<b>Electroválvula SIRAI L172V03 DC/CC</b>	4

Tabla 3: Cantidad de componentes actuadores y sensores

<b>DISEÑO PCB</b>	
<b>Nombre</b>	<b>Cantidad</b>
<b>Condensador 1 <math>\mu</math>F (399-4873-1-ND)</b>	4
<b>Resistencia 165<math>\Omega</math> (P165DATR-ND)</b>	8
<b>Resistencia 1k<math>\Omega</math> (311-1.00KLRCT-ND)</b>	4
<b>Amplificador doble (296-7148-1-ND)</b>	2
<b>Relé 24DC (JS1-5V-F)</b>	4
<b>Resistencia 300<math>\Omega</math> (ERA-2AEB301X)</b>	8
<b>Resistencia 10k<math>\Omega</math> (311-10.0KLRCT-ND)</b>	4

<b>Transistor BJT (MMBT3904-FDICT-ND)</b>	4
<b>Resistencia 85Ω (Y162485R0000B9R)</b>	4
<b>Jumper (QPC02SXGN-RC)</b>	4
<b>Pulsador Reset (450-1650-ND)</b>	1
<b>Bloque terminal 10 (277-1255-ND)</b>	2
<b>Bloque terminal 2 (1751248)</b>	1
<b>SODIAL(R) 1X 40 Pin 2.54mm Male PCB</b>	1

Tabla 4: Cantidad de componentes para el diseño de PCB





# **PRESUPUESTO**



<b>Componentes sensores y actuadores</b>		
<b>Nombre</b>	<b>Cantidad</b>	<b>Precio</b>
<b>Regulador 840 &amp; 830 Side-Trak® Analog Mass Flow Controller/Meter</b>	2	2668
<b>Placa Arduino Due</b>	1	44,71
<b>SIL sensor presión 40 bar</b>	2	51,42
<b>Sensor de caudal SFAH</b>	2	896,28
<b>Electroválvula SIRAI L172V03 DC/CC</b>	4	174,24
<b>TOTAL</b>		3834,65

Tabla 5: Presupuesto componentes actuadores y sensores

<b>DISEÑO PCB</b>		
<b>Nombre</b>	<b>Cantidad</b>	<b>Precio</b>
<b>Condensador 1 <math>\mu</math>F (399-4873-1-ND)</b>	4	0,36
<b>Resistencia 165<math>\Omega</math> (P165DATR-ND)</b>	8	2,24
<b>Resistencia 1k<math>\Omega</math> (311-1.00KLRCT-ND)</b>	4	0,32
<b>Amplificador doble (296-7148-1-ND)</b>	2	0,96
<b>Relé 24DC (JS1-5V-F)</b>	4	5,52
<b>Resistencia 300<math>\Omega</math> (ERA-2AEB301X)</b>	8	2,96

<b>Resistencia 10k<math>\Omega</math> (311-10.0KLRCT-ND)</b>	4	0,32
<b>Transistor BJT (MMBT3904-FDICT-ND)</b>	4	0,4
<b>Resistencia 85<math>\Omega</math> (Y162485R0000B9R)</b>	4	1
<b>Jumper (QPC02SXGN-RC)</b>	4	0,32
<b>Pulsador Reset (450-1650-ND)</b>	1	0,08
<b>Bloque terminal 10 (277-1255-ND)</b>	2	8,98
<b>Bloque terminal 2 (1751248)</b>	1	1,1
<b>SODIAL(R) 1X 40 Pin 2.54mm Male PCB</b>	1	1,67
<b>TOTAL</b>		26,23

Tabla 6: Presupuesto componentes PCB

<b>Fabricación PCB en PCBA store online</b>	
<b>precio</b>	34

Tabla 7: Presupuesto fabricación PCB

<b>Licencia Codesys Version: 3.5.16.10</b>	
<b>precio</b>	504,00

Tabla 8: Presupuesto licencia Codesys

<b>Trabajo de ingeniería</b>			
<b>Trabajo</b>	<b>horas</b>	<b>precio/hora</b>	<b>total</b>
<b>Diseño PCB</b>	84	30.00	2520,00

<b>Programación Arduino</b>	27	30.00	810,00
<b>Programación Codesys</b>	39	30.00	1170,00
<b>TOTAL</b>			4500,00

Tabla 9: Presupuesto trabajo ingenieril

<b>Presupuesto ejecución material</b>	
<b>Concepto</b>	<b>Coste</b>
<b>Componentes sensores y act</b>	3834,65
<b>Diseño PCB</b>	26,23
<b>Fabricación PCB</b>	34,00
<b>Licencia codesys</b>	504,00
<b>Trabajo de ingeniería</b>	4500,00
<b>TOTAL</b>	8898,88

Tabla 10: Presupuesto ejecución material

<b>Presupuesto de ejecución por contrata parcial</b>	
<b>Concepto</b>	<b>Coste</b>
<b>Presupuesto ejecución material</b>	8898,88
<b>Gastos generales(15%)</b>	1334,83
<b>Beneficio industrial(6%)</b>	533,93
<b>TOTAL</b>	10767,64

Tabla 11: Presupuesto ejecución por contrata parcial

*Agradecimientos: debería de redactar otro TFG sobre toda la ayuda y apoyo que he recibido a lo largo de mi carrera. Espero que mi más sincero agradecimiento a mis amigos, familia, pareja, profesores y compañeros de clase sea suficiente, por estar en los mejores y peores momentos y hacerme crecer como persona. MUCHAS GRACIAS!!!*