



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Barcelona Est

TRABAJO DE FINAL DE GRADO

Grado en Ingeniería electrónica industrial y automática

**DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA IOT
PARA UNA PLANTA PILOTO DE OBTENCIÓN DE H2
MEDIANTE REFORMADORES DE MEMBRANA**



Memoria y Anexos

Autor/a: Daniel Noguera Segura

Director/a: Javier Francisco Gámiz Caro

Co-Director/a: Ricardo Torres Cámara

Convocatoria: Mayo 2021

Índice

Índice	3
Resumen.	1
Resum.	2
Abstract.	3
1. Introducción.	4
1.1. Objetivos.	4
1.2. Alcance.	4
1.3. Motivación.	5
1.4. Estructura de la memoria.	5
2. Análisis del problema.	7
2.1. Descripción del proceso.	7
2.1.1. Introducción.	7
2.1.2. Química del proceso.	7
2.1.3. Descripción de la planta real.	8
2.1.4. Modelo matemático del proceso.	10
2.2. Sistema de control.	11
2.2.1. Equipos.	11
2.2.2. Fases del proceso.	13
2.3. Requerimientos funcionales.	14
2.4. Planificación de los trabajos.	17
3. Diseño e implementación de la solución.	18
3.1. Arquitectura del sistema.	18
3.1.1. Hardware del sistema	18
3.1.2. Software del sistema	18
3.1.2.1. U-create web	18
3.1.2.2. Microsoft Azure.	19
3.1.2.3. Power BI	19
3.1.3. Vista global de comunicaciones.	20
3.2. Simulación del proceso	21
3.2.1. Datos recibidos.	22
3.2.2. Simulación	23
3.2.3. Envío de datos.	25
3.3. Programación del sistema de control.	27
3.3.1. Real Time Application.	27
3.3.1.1. Definición de los tipos de datos.	27

3.3.1.2. Programa	30
3.3.2. Node-red.	33
3.4. Creación del panel IoT.	36
3.4.1. Envío de datos a Azure.	38
3.4.2. Azure + Power BI.	39
4. Pruebas y resultados	43
4.1. Comunicación entre el PLC y el Simulink.	43
4.2. Envío de datos al centro IoT de Azure.	43
4.3. Comunicación entre la RTA y el Node-Red	43
4.4. Sintonización de los PIDs.	44
5. Conclusiones.	45
6. Futuras líneas de trabajo.	46
7. Webgrafía.	47
Anexo I: Programación.	49
Anexo II: Equipos y tarjetas.	50
Anexo III: Memoria económica.	51

Resumen.

El objetivo principal de este proyecto es diseñar e implementar una plataforma IoT para una planta piloto de obtención de hidrógeno. Para lograr este objetivo, previamente se ha tenido que crear un modelo simulado de dicha planta y un sistema de control capaz de controlarla.

El modelo simulado se ha llevado a cabo mediante *Simulink*. Esta simulación es controlada por una Interfaz Gráfica de Usuario que también permite el control de las válvulas de la planta.

El sistema de control ha sido programado con el PLC *UC20-WL2000-AC*, con unos algoritmos que permiten la elección de dos modos de operación: manual o automático. Con la ayuda de este PLC, que también hace de pasarela IoT, se ha logrado crear un panel IoT que permite monitorizar todas las variables del sistema en tiempo real.

En este documento se explica detalladamente cómo se ha obtenido la solución final. En el anexo I se muestra toda la programación realizada; en el anexo II se detallan las tarjetas de entrada/salida del PLC compatibles con los equipos de la planta real, y como conectarlos entre ellos y, finalmente, en el anexo III se hace un análisis económico del proyecto.

Resum.

L'objectiu principal d'aquest projecte és dissenyar i implementar una plataforma IoT per a una planta pilot d'obtenció d'hidrogen. Per aconseguir aquest objectiu, prèviament s'ha hagut de crear un model simulat d'aquesta planta i un sistema de control capaç de controlar-la.

El model simulat s'ha dut a terme mitjançant Simulink. Aquesta simulació és controlada per una Interfície Gràfica d'Usuari que també permet el control de les vàlvules de la planta.

El sistema de control ha estat programat amb el PLC *UC20-WL2000-AC*, amb uns algoritmes que permeten l'elecció de dos modes d'operació: manual o automàtic. Amb l'ajuda d'aquest PLC, que també fa de passarel·la IoT, s'ha aconseguit crear un panell IoT que permet monitoritzar totes les variables de sistema en temps real.

En aquest document s'explica detalladament com s'ha obtingut la solució final. A l'annex I es mostra tota la programació realitzada; a l'annex II es detallen les targetes d'entrada/sortida del PLC compatibles amb els equips de la planta real, i com connectar-los entre ells i, finalment, a l'annex III es fa un anàlisi econòmic del projecte.

Abstract.

The main objective of this project is to design and implement an IoT platform for a pilot plant for obtaining hydrogen. To achieve this objective, a simulated model of this plant and a control system capable of controlling it has been previously created.

The simulated model has been done with Simulink. This simulation is controlled by a Graphical User Interface that also allows the control of the plant valves.

The control system has been programmed with the PLC *UC20-WL2000-AC*, with algorithms that allow the choice of two modes of operation: manual or automatic. With the help of this PLC, which also acts as an IoT gateway, it has been possible to create an IoT dashboard that allows all the variables of the system to be monitored in real time.

This document explains in detail how the final solution has been obtained. Finally, in annex I it's shown all the programming done, in annex II the PLC input/output cards compatible with the equipment of the real plant are detailed, and how to connect them, and finally, in annex III is made an economic analysis of the project.

1. Introducción.

1.1. Objetivos.

El objetivo de este proyecto es diseñar e implementar una plataforma IoT para una planta piloto de obtención de hidrogeno. La solución final debe de mostrar en tiempo real en un panel las variables del sistema, el estado del sistema de control de la planta (en marcha o parado) y su modo de operación (automático o manual). Este panel debe de ser accesible desde cualquier *smartphone*, tableta u ordenador.

Esto de forma indirecta implica otro objetivo, que es el de implementar el sistema de control de la planta.

1.2. Alcance.

El alcance de este proyecto se divide en las siguientes actividades:

- Estudiar el sistema con el objetivo de tener claro el comportamiento de las variables involucradas y el funcionamiento de los sensores y actuadores utilizados.
- Crear un modelo matemático que describa el comportamiento de la planta real e implementarlo en un modelo simulado.
- Crear una Interfaz Gráfica de Usuario mediante *GUIDE* de *Matlab* que permita controlar el inicio y paro de la simulación y las válvulas del sistema, además de mostrar los valores de los sensores en tiempo real.
- Crear un sistema de control que actúe sobre las variables del sistema y poder controlarlo con una interfaz que permita también monitorizar los valores medidos por los sensores.
- Crear un sistema capaz de almacenar localmente en diferentes archivos los datos recogidos por los sensores y poder visualizarlos gráficamente en cualquier momento.
- Crear un panel de control que permite monitorizar las variables de la planta y el estado del sistema de control mediante tecnología IoT.

1.3. Motivación.

La motivación a realizar este proyecto surge del interés en el mundo de la automatización industrial durante el segundo cuatrimestre de 2019, en el que cursé la asignatura *Implementación de Sistemas Automáticos (ISA)*. Me pareció una rama de la ingeniería muy interesante. En este proyecto vi la oportunidad de adentrarme en ella y seguir aprendiendo. A demás, me llamo la atención la palabra *IoT* en el título, de la cual ya había oído hablar alguna vez, pero no tenía muy claro que era realmente.

Por otro lado, en el ámbito personal este proyecto me suponía una oportunidad para aplicar de forma práctica todos los conocimientos adquiridos a lo largo del curso; aprender a trabajar en un proyecto; aprender a ser resiliente ante los problemas que se van planteando y a encontrar soluciones buenas y eficientes.

1.4. Estructura de la memoria.

El documento está estructurado de la siguiente forma:

Apartado 2: Análisis del problema. Se hace una descripción del proceso a automatizar, donde se hace una breve explicación del método de producción de hidrogeno utilizado, se muestran las reacciones químicas que intervienen y como se llevan a cabo dentro de la planta. También se da el modelo matemático de la planta. Posteriormente, se habla de los equipos que conforman el sistema de control y las fases del proceso. Finalmente, se detallan los requerimientos funcionales y se hace una planificación de los trabajos.

Apartado 3: Diseño e implementación de la solución. Inicialmente, se habla sobre la arquitectura del sistema, donde se detalla el hardware y software utilizado y se da una vista global de las comunicaciones. Se explica cómo se ha creado el modelo simulado, la programación del sistema de control y, finalmente, se detalla cómo se ha llevado a cabo el diseño e implementación de la plataforma IoT.

Apartado 4: Pruebas y resultados. Se detallan las pruebas llevadas a cabo durante el proyecto.

Apartado 5: Conclusiones.

Apartado 6: Futuras líneas de trabajo.

Apartado 7: Webgrafía.

Anexo I: Programación. Se muestra todo el programa que se ha realizado para desarrollar el proyecto.

Anexo II: Equipos y tarjetas. Se presentan las tarjetas de entrada/salida del PLC compatibles con los equipos del sistema, y se explica cómo comunicarlas entre ellas.

Anexo III: Memoria económica. Se da una idea de lo que podría costar implementar el proyecto.

2. Análisis del problema.

Se quiere diseñar e implementar una plataforma IoT que monitorice las variables de la planta. Como no es posible acceder a la planta real del laboratorio, puesto que está siendo utilizada por otro proyectista, se debe de crear un modelo simulado. También se ha de crear un sistema de control que nos permita controlar las variables del sistema, en función de lo que requiera el proceso.

2.1. Descripción del proceso.

2.1.1. Introducción.

El proceso a automatizar tiene como fin la obtención de hidrogeno. Existen varios métodos de obtención de hidrogeno, pero en este proceso se hace mediante reactores catalíticos de membranas. Este consiste en producir y separar el hidrogeno dentro del mismo reactor donde se produce la reacción, de manera simultánea, consiguiendo así una pureza más alta. En otros procesos esto no es así, pues la separación se produce en una fase posterior.

Dentro del reactor se introduce una membrana normalmente hecha de paladio o aleaciones de este material. El motivo por el cual se suele usar este material es su alta selectividad al hidrogeno. A la vez que se produce la reacción, esta membrana separa el hidrogeno, mediante un gradiente de presión.

El proceso requiere de una temperatura de operación superior a los 300 °C. Esto se debe a que a temperaturas inferiores se puede producir fracturación mecánica de la membrana, por la formación de hidruro de paladio. Este proceso se conoce como fragilización por hidrogeno.

2.1.2. Química del proceso.

El hidrógeno se obtiene a partir de una mezcla de metanol y agua. Las reacciones químicas que se producen dentro del reactor son las siguientes:



Las cantidades producidas de cada compuesto dependen, principalmente, de la cantidad de metanol y agua, y de la temperatura y presión del reactor.

2.1.3. Descripción de la planta real.

La planta piloto del laboratorio es la siguiente:

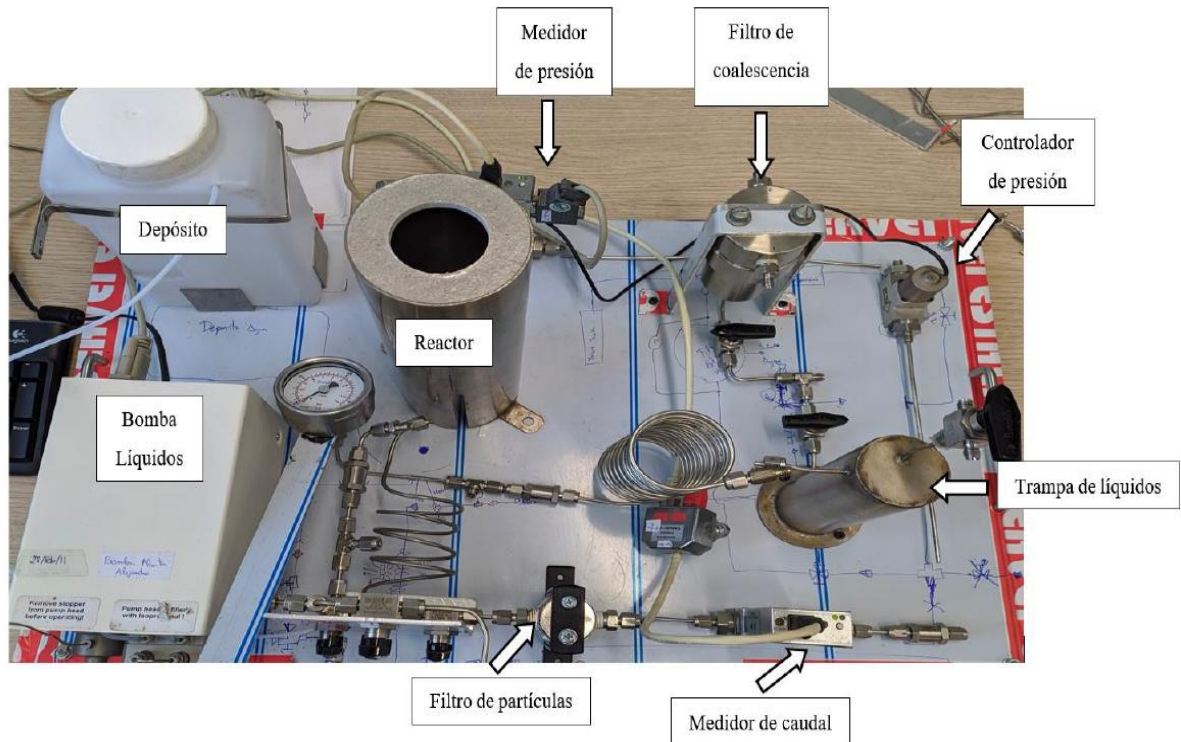


Ilustración 1: Planta piloto del laboratorio (Imagen obtenida del documento de presentación del proyecto del doctorando que trabajaba con la planta).

La siguiente imagen muestra un diagrama PI&D de la planta:

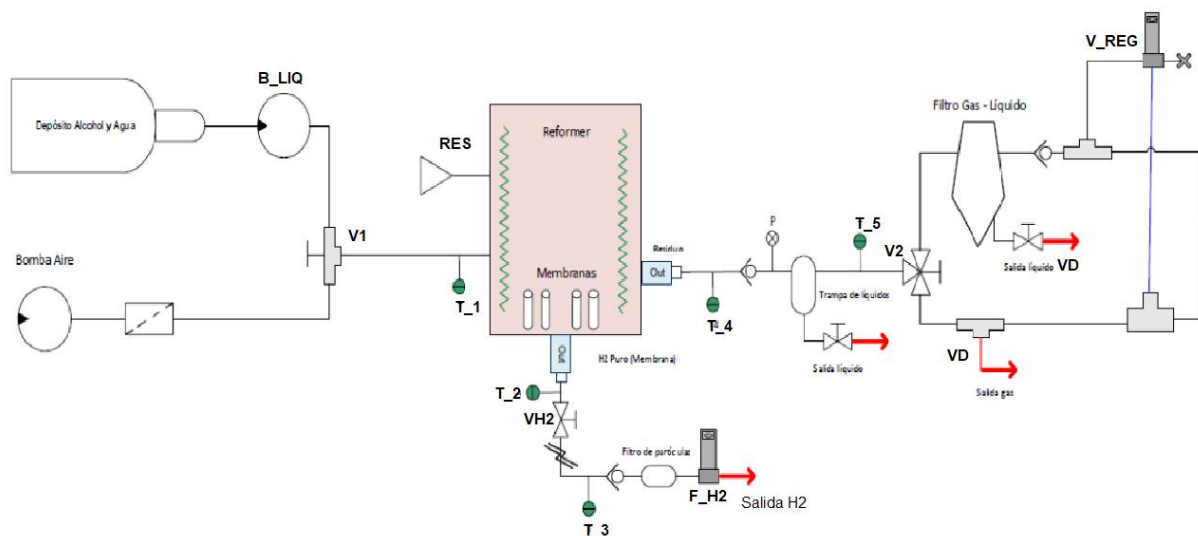


Ilustración 2: PI&D de la planta.

A continuación, se muestran los elementos mecánicos que forman parte del sistema:

NOMBRE	NOMBRE CODIFICADO	ESTADOS
Válvula de tres vías 1	V1	ESTADO 1 = circula metanol ESTADO 2 = circula nitrógeno
Válvula de tres vías 2	V2	ESTADO 1 = presión controlada ESTADO 2 = presión atmosférica
Válvula de H ₂	Vh2	ABIERTA = pasa H ₂ CERRADA = no pasa H ₂
Válvulas de desagüe	Vd	ABIERTA = circula gas CERRADA = no circula gas

Tabla 1: Elementos mecánicos del sistema.

La planta está compuesta por un depósito donde se almacena la mezcla de metanol y agua. Una vez se acciona la bomba de líquidos, la mezcla va hacia el reformador, donde se lleva a cabo la reacción química. Una resistencia eléctrica, controlada por un controlador de temperatura, calienta el reformador. Este tiene dos salidas, una para el hidrogeno y otra para el residuo.

El residuo va directo hacia una trampa de líquidos donde parte de este pasa a estado líquido. A continuación si la válvula de tres vías 2 (V2) está en estado 1, el residuo restante que no se haya condensado pasa por un filtro para, posteriormente, ir directo a la atmosfera. En esta misma salida hay una válvula reguladora que es la que permite obtener las variaciones de presión dentro del reactor.

El hidrogeno formado pasa por la válvula Vh2, que está abierta durante el funcionamiento. Posteriormente se encuentra con un filtro de partículas que quita las impurezas que pudieran quedar. Finalmente, el hidrogeno entra en la pila de combustible o el depósito donde se quiera almacenar. El caudal de hidrogeno es medido por un caudalímetro que se encuentra inmediatamente después del filtro de partículas.

Además, el sistema tiene repartidos un total de seis termopares que nos permiten conocer la temperatura en diferentes puntos del sistema.

Cada vez que se finaliza el proceso de formación de hidrógeno, se pone en estado 2 la válvula de tres vías 1 (V1) y se activa manualmente la bomba de aire. Esta hace pasar

nitrógeno para limpiar todo el sistema. Es necesario llevar a cabo este proceso de limpieza cada vez que se finaliza el proceso de obtención de hidrogeno

2.1.4. Modelo matemático del proceso.

Para simular el proceso es necesario tener el modelo matemático que defina el comportamiento de las variables del sistema en el reactor. Debido a que no se dispone de un modelo matemático que describa la planta, se ha creado uno basado en modelos con comportamientos similares. Para crear un modelo simulado sencillo y lo más práctico posible, se utiliza una función de transferencia de primer orden como modelo matemático para las dos variables que participan en el sistema (temperatura y presión). Para simular el caudal de salida de hidrogeno, en cambio, se utilizan unas tablas que indican el caudal de hidrogeno producido en función de la temperatura y presión del reactor, a un caudal de entrada determinado.

TEMPERATURA:

El modelo matemático de un sistema térmico cerrado puede aproximarse al que describe una función de transferencia de primer orden con retraso. El modelo matemático que describe el comportamiento de la temperatura en el reactor de la planta se describe con la siguiente función de transferencia:

$$G(s) = \frac{2.77e^{-12s}}{861s + 1} \quad (4)$$

Se ha utilizado una función de transferencia con una constante de tiempo elevada, al igual que un tiempo de retraso alto, típico de un sistema térmico, debido a la inercia térmica del sistema.

PRESIÓN:

La presión también se puede modelar con una función de transferencia de primer orden, pero sin retraso y con una constante de tiempo considerablemente inferior al de la temperatura. Esto se debe a que el efecto de la variación de la posición de la válvula afecta de una forma más inmediata sobre la presión del sistema. El modelo matemático que describe el comportamiento de la presión en el reactor se describe con la siguiente función de transferencia:

$$G(s) = \frac{1}{150s + 1} \quad (5)$$

HIDRÓGENO:

La cantidad de hidrogeno que produce el sistema depende del caudal de entrada, la temperatura y la presión del reactor. A continuación, se muestran dos tablas que recogen los valores aproximados del caudal de hidrogeno producido según la temperatura y presión del reactor. La primera tabla es a un caudal de entrada de 0.02 ml/min y la segunda a un caudal de entrada de 0.2 ml/min. Solo son necesarias las tablas a estos dos caudales de entrada puesto que son los únicos valores con los que se va a trabajar.

T(°C)/P(bar)	1	2	4	6	8	10	12
450	0.12	0.23	0.45	0.66	0.87	0.99	1.20
500	0.20	0.29	0.47	0.65	0.83	1.00	1.20
550	0.30	0.38	0.54	0.71	0.87	1.03	1.20
600	0.40	0.47	0.62	0.76	0.90	1.12	1.20

Tabla 2: Caudal de hidrógeno producido (en Nml/min) en función de la temperatura y presión, a 0.02 ml/min de entrada de metanol y agua.

T(°C)/P(bar)	1	2	4	6	8	10	12
450	12.00	23.00	45.00	66.00	87.00	99.00	120.00
500	20.00	29.09	47.27	65.45	83.64	101.82	120.00
550	30.00	38.17	54.54	70.90	87.26	103.63	120.00
600	40.00	47.273	61.818	76.36	90.91	105.45	120.00

Tabla 3: Caudal de hidrógeno producido (en Nml/min) en función de la temperatura y presión, a 0.2 ml/min de entrada de metanol y agua.

2.2. Sistema de control.

El sistema de control tiene como objetivo controlar la temperatura y presión del reformador, y el caudal de metanol y agua que entra en este. Antes de programar el sistema, es necesario saber los equipos que se disponen y las fases del proceso.

2.2.1. Equipos.

La siguiente tabla muestra el equipo de sensores y controladores que tiene la planta real.

EQUIPO	NOMBRE	MARCA	FUNCIÓN
Bomba de líquidos	Smartline Pump 100	Knauer	Actuador
Controlador de temperatura	Piccolo Controller P116	Eurotherm	Actuador/sensor
Controlador de presión	EL-PRESS	Bronkshort	Actuador/sensor
Caudalímetro	LOW- Δ P-FLOW	Bronkshort	Sensor
Tarjeta de lectura de termopares	PM8TCISO	Promux	Sensor

Tabla 4: Controladores y sensores de la planta.

(En el Anexo II se entra en detalle sobre los equipos utilizados. También se mencionan las tarjetas de entrada/salida de Weidmüller compatibles con estos equipos, y como conectarlos entre ellos para lograr una comunicación. Para la solución que se quiere obtener en este proyecto, esto no tiene importancia, pero se explica en el anexo por que ha formado parte del aprendizaje del proyecto. Además, puede ser muy útil a quien en un futuro quisiera implementar la solución obtenida en este proyecto a la planta real.

Tanto el controlador de temperatura como el de presión incorporan un PID propio, lo que hace que no haya que implementar un sistema de control PID en nuestro PLC.

Para la implementación de la solución en el modelo simulado, es necesario tener claro cuáles son las señales de entrada y salida. A continuación, se muestra una tabla de todos los dispositivos, junto a sus señales de entrada y salida:

NOMBRE	NOMBRE CODIFICADO	SEÑALES	DESCRIPCIÓN	TIPO
Bomba de líquidos	B_LIQ	C_FIN	Consigna caudal	SA
		M_B	Salida marcha bomba	SD
Controlador de presión	V_REG	C_P	Consigna de presión	SA
		EA_P2	Entrada presión	EA
Caudalímetro	F_H2	EA_FH2	Entrada caudal H2	EA
Controlador de temperatura	RES	EA_T	Temperatura reactor	EA
		C_T	Consigna temperatura	SA
Tarjeta de lectura de termopares	T_1	EA_T1	Temperatura TP1	EA
	T_2	EA_T2	Temperatura TP2	EA
	T_3	EA_T3	Temperatura TP3	EA
	T_4	EA_T4	Temperatura TP4	EA
	T_5	EA_T5	Temperatura TP5	EA

Tabla 5: Elementos y variables del sistema.

2.2.2. Fases del proceso.

El proceso se divide en 4 fases. Estas son:

- Preparación (fase 0): Fase de preparación de la planta que se lleva a cabo antes de poner en marcha el sistema de control.
- Arranque (fase 1): Fase en la que se pone en marcha el sistema de control.
- Funcionamiento (fase 2): Fase en la que se obtiene el hidrogeno.
- Parada (fase 3): Fase en la que se para el sistema.

PREPARACION DE LA PLANTA (fase 0):

Esta fase es la previa a la del inicio del control y, por tanto, no se ejecuta automáticamente, sino que es el operario quien se encarga de llevarla a cabo.

Consiste en: asegurarse de que el depósito contiene una cantidad considerable de la mezcla de agua y metanol para llevar a cabo el proceso; poner la válvula de tres vías 1 (V1) de modo que permita el paso de la mezcla; cerrar las válvulas de desagüe (Vd) y de hidrógeno (Vh2) y poner la válvula de tres vías 2 (V2) de modo que el circuito de presión quede cerrado.

ARRANQUE (fase 1):

La fase de arranque consiste en poner en marcha la planta. Inicialmente se ha de alcanzar una temperatura en el reactor de 450 °C. Una vez alcanzada esta temperatura, se introduce en el reactor un pequeño caudal de metanol y agua a modo de prueba. A continuación, el operario debe de abrir la válvula de hidrogeno para comprobar que efectivamente se está produciendo hidrogeno.

FUNCIONAMIENTO (fase 2):

Una vez se ha cumplido el protocolo de arranque, se lleva a cabo esta fase, en la que el sistema empieza a producir las cantidades de hidrogeno deseadas.

En este punto, lo principal para que la planta empiece a producir cantidades de hidrogeno más elevadas, es aumentar el caudal de metanol y agua que entra en el reactor. Por lo tanto, en esta fase se introduce un caudal de entrada diez veces superior al de la fase de arranque. En este punto, el operario juega con las variaciones

de temperatura y presión del sistema, según el caudal de hidrogeno que desee obtener.

PARADA (fase 3):

Esta fase es la que entra en juego una vez ha finalizado el proceso de obtención de hidrógeno. La parada tiene que seguir un protocolo puesto que, si se hiciese de manera descontrolada, podría dañarse la planta.

El protocolo de parada empieza poniendo el reactor a presión atmosférica. Una vez el reactor se encuentra a presión atmosférica, se deja de introducir metanol y agua en este y, finalmente, se disminuye la temperatura del reactor lentamente hasta alcanzar la temperatura ambiente.

2.3. Requerimientos funcionales.

A continuación, se detallan los requerimientos funcionales del sistema:

RQ1	La interfaz de control tiene que tener una pantalla de inicio donde se muestre un dibujo de la planta y se vean los valores actuales y las consignas. También tiene que tener un apartado donde se muestren mensajes.
RQ2	La interfaz de control tiene que tener una pantalla de visualización donde se muestre mediante gráficos lineales la temperatura y presión del reactor; el caudal de entrada de metanol y agua en el depósito y el caudal de salida de hidrogeno.
RQ3	La interfaz de control tiene que tener una pantalla de control donde se puedan seleccionar el modo de operación (automático/manual) y las consignas del sistema. RQ4: Antes de iniciar el proceso, deben de aparecer una serie de mensajes en la interfaz de control que recuerden los pasos a seguir en el protocolo de preparación de la planta.
RQ5	No se puede iniciar el proceso de control hasta que no se termine el protocolo de preparación.
RQ6	La pulsación de un botón de <i>INICIO</i> , inicia el proceso, consignando la temperatura a 450 °C.

RQ7	Una vez la temperatura llega al valor de consigna, se consigna un caudal de entrada de 0.02 ml/min a la bomba de líquidos. En este momento, el operario abre la válvula de hidrogeno y comprueba que se produce hidrogeno.
RQ8	Cuando se selecciona el modo de operación, se consigna un caudal de entrada a la bomba de líquidos de 0.2 ml/min.
RQ9	El sistema debe de permitir el cambio de modo de operación en cualquier momento.
RQ10	Las variables del sistema nunca pueden salir de los siguientes rangos de valores, una vez el sistema se encuentra en fase de funcionamiento: temperatura: 450 – 600 °C; presión: 1 – 12 bar; caudal de hidrogeno: 0 – 120 NI/min.
RQ11	En el modo de operación manual, el sistema recibe las consignas de temperatura y presión.
RQ12	En el modo manual, los valores de consigna que se pueden seleccionar van desde 450 °C a 600 °C, de 50 en 50.
RQ13	Si el sistema recibe una consigna de presión superior a 12 bares, aparece un mensaje en la pantalla que indica que el valor es demasiado alto.
RQ14	En cualquiera de los modos de operación, la presión no empieza a variar hasta que la temperatura no haya alcanzado su valor de consigna.
RQ15	En el modo automático, la temperatura de operación es siempre de 450 °C.
RQ16	Cuando se selecciona el modo de operación automático, se consigna un valor de temperatura de 450 °C automáticamente.
RQ17	En el modo de operación automático, el operario introduce el valor de hidrogeno que desea obtener y el sistema calcula la consigna de presión necesaria a 450 °C para alcanzar ese valor de hidrogeno.
RQ18	En el modo de operación automático, si el operario introduce la consigna de hidrogeno antes de que la temperatura haya alcanzado los 450 °C, aparece un mensaje que informa al operario de que debe de esperar a que la temperatura alcance los 450 °C.
RQ19	En el modo de operación automático, si el operario introduce un valor de

	consigna de hidrogeno superior a 120 Nml/min, debe de aparecer un mensaje que informe al operario que el valor es demasiado alto.
RQ20	Desde que se inicia el proceso hasta que se para, el sistema debe de guardar los valores de temperatura y presión del reactor e hidrogeno de salida en un archivo donde se indique en su propio nombre el inicio del proceso. Estos valores se deben de poder visualizar en cualquier momento en una gráfica lineal.
RQ21	Se podrá observar en un panel IoT los valores de temperatura y presión del reactor; el caudal de hidrogeno de salida y los estados (inicio/paro y manual/automático) del sistema de control en tiempo real. Este panel será accesible desde cualquier <i>smartphone</i> , tableta u ordenador y en cualquier momento.
RQ22	El sistema iniciará el protocolo de parada cuando se pulse el botón de paro de la interfaz de control o la parada de emergencia del propio <i>Simulink</i> .
RQ23	Cada vez que la planta se encuentre en el proceso de parada, se recordará en el apartado de mensajes de la interfaz de control el protocolo de parada a seguir por el operario.
RQ24	El protocolo de parada empieza con una consigna de presión de 1 bar. Cuando el sistema alcanza este valor, el operario debe de poner la válvula v2 de modo que el sistema se encuentre a presión atmosférica.
RQ25	En el protocolo de parada, cuando la presión del sistema haya alcanzado 1 bar, se ha de parar la bomba de líquidos.
RQ26	En el protocolo de parada, cuando la bomba de líquidos se haya parado, se consigna un valor de temperatura de 25 °C.
RQ27	Una vez el protocolo de parada ha terminado y se ha alcanzado la temperatura de 25 °C, el sistema de control debe de poder volver a iniciar el proceso.

2.4. Planificación de los trabajos.

A continuación, se muestran las tareas a realizar para desarrollar el proyecto de forma ordenada:

- Estudio de la planta y de sus equipos.
- Creación del modelo simulado.
- Creación del sistema de control.
- Implementación de la solución de la plataforma IoT.
- Memoria técnica.

3. Diseño e implementación de la solución.

3.1. Arquitectura del sistema.

Es muy importante tener claro desde un inicio la arquitectura de la solución que se quiere implementar. En este apartado se habla de los equipos utilizados para implementar la solución y el *software* utilizado. También hace una vista global de las comunicaciones.

3.1.1. Hardware del sistema

La siguiente tabla muestra los dispositivos utilizados:

EQUIPO	MARCA	MODELO
PLC	Weidmüller	UC20-WL-2000-AC
ROUTER	ZTE	ZXHN F680
Ordenador	Acer	Aspire E5-575
Tableta	Samsung Galaxy	Tab A
Servidores de Azure	NA	NA
Tarjeta microSD	Kingston	SDCS2/32GB

Tabla 6: Hardware del sistema.

El PLC juega el papel principal de la solución, puesto que es donde se encuentra el programa que controla el sistema y su interfaz de control. Este dispone de conexión a internet si se conecta mediante un cable Ethernet desde uno de sus puertos RJ-45 a un rúter (en este caso, ZTE ZXHN F680). Gracias a esto, el PLC puede hacer la función de pasarela IoT, permitiendo así el envío de datos a los servidores de *Microsoft Azure*. Estos, posteriormente, serán visualizados en la tableta. A demás, el PLC incorpora una microSD de 32 GB para guardar los datos medidos por los sensores.

La función del ordenador es poner en marcha la simulación y manejar la interfaz de control del PLC. Es también en el ordenador donde se visualizan los datos guardados en la microSD del PLC en forma de gráficos lineales.

3.1.2. Software del sistema

3.1.2.1. U-create web

El software utilizado para programar el controlador es el *u-create web*. Se trata de un *software* que viene incorporado dentro del propio controlador. Se accede introduciendo la dirección IP del controlador en el navegador web. Este incorpora dos

aplicaciones que nos permiten programar el control: la *Aplicación en Tiempo Real (RTA)* y el *Node-Red*.

La *RTA* es la que controla los dispositivos conectados al controlador y maneja los datos del proceso en tiempo real, tal y como indica su propio nombre. Está programada de acuerdo con el estándar IEC 61131-3, mediante el lenguaje gráfico de diagrama de bloques de funciones. Es aquí donde se programa la parte lógica del programa, además de su interfaz de control. Dispone de variables globales, que permiten el intercambio de datos con el *Node-Red*.

Por otro lado, *Node-Red* es una herramienta de programación visual que permite al usuario programar sin escribir código, añadiendo nodos y conectándolos entre ellos. Hay una gran cantidad de nodos con diferentes funcionalidades y, por supuesto, da la posibilidad de crear funciones propias mediante JavaScript. En definitiva, es una herramienta muy visual y sencilla de utilizar que ofrece muchas posibilidades al ámbito industrial. Gracias a *Node-Red* es posible implementar el panel IoT.

3.1.2.2. *Microsoft Azure.*

Microsoft Azure nos ofrece dos herramientas fundamentales para la creación del sistema IoT. *Azure IoT Hub* es el servicio que nos permite la comunicación bidireccional entre nuestro PLC y nuestra aplicación IoT. Por otro lado, *Stream Analytics Job* es el que se encarga de procesar los datos que recibe *Azure IoT Hub* en tiempo real y mandarlos como salida a *Power BI*.

3.1.2.3. *Power BI*

Power BI es una herramienta de análisis de datos. Es la herramienta con la que se crea el panel donde se visualizan en tiempo real los datos enviados al Centro de IoT de *Microsoft Azure*.

3.1.3. Vista global de comunicaciones.

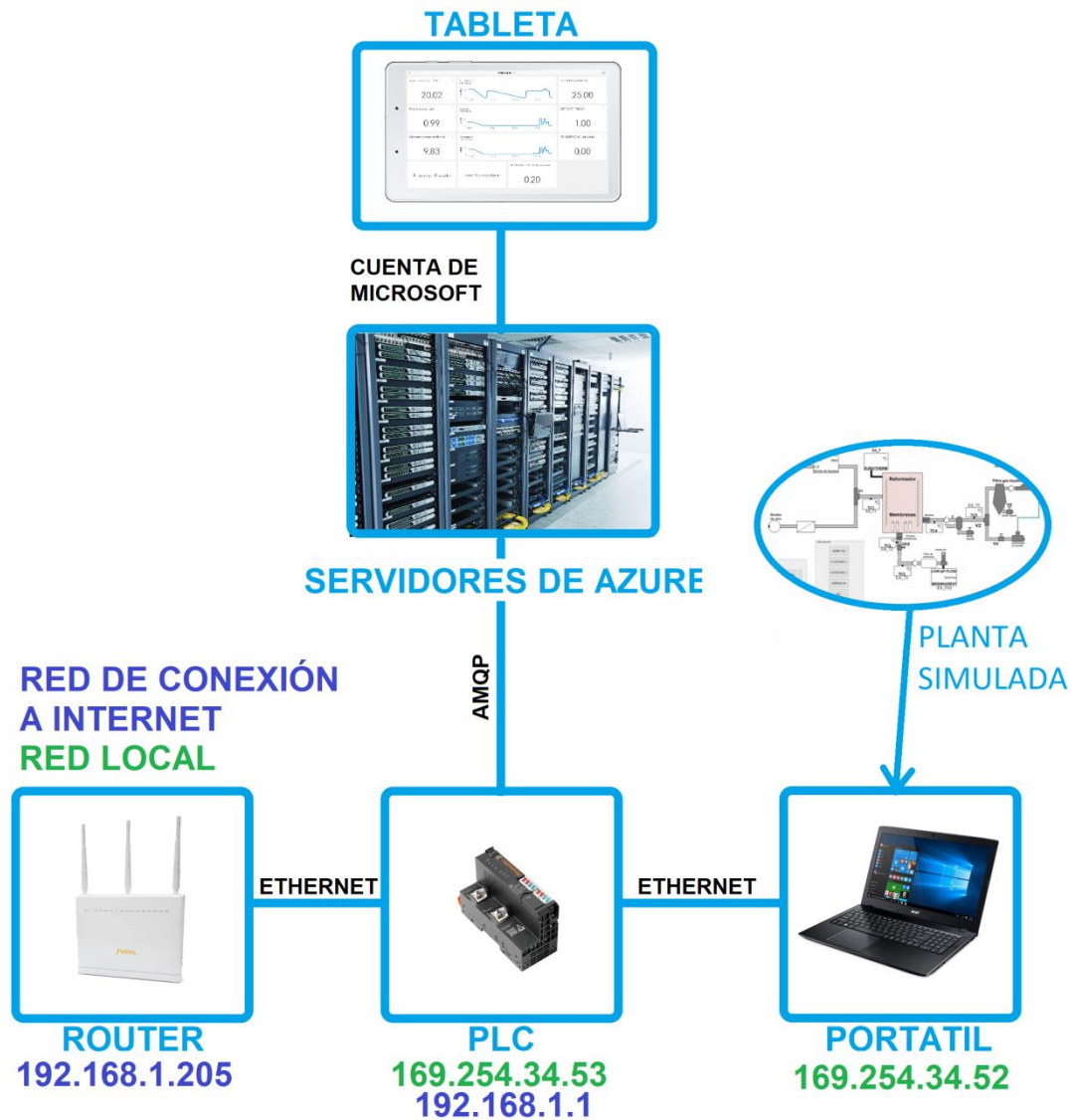


Ilustración 3: Vista global de las comunicaciones.

El ordenador portátil y el PLC están conectados entre ellos mediante un cable *Ethernet*. Esta conexión permite el intercambio de datos entre el *Simulink* y el PLC, mediante el protocolo UDP. Es un protocolo que nos permite transmitir datagramas de forma rápida en redes IP. A pesar de no ser tan seguro como el protocolo TCP/IP en cuanto a la fiabilidad de llegada de datos, estos se transmiten de una forma más rápida y requieren de menos procesamiento de la CPU.

Los datos enviados desde el PLC hasta los servidores de *Microsoft Azure* son enviados mediante el protocolo *AMQP*. Es un protocolo que proporciona una comunicación segura entre nuestra fuente de datos (el PLC) y la aplicación IoT. El uso de este protocolo solo tiene sentido cuando existe un intermediario, en este caso, el Centro IoT de *Microsoft Azure*.

Finalmente, la comunicación entre *Power Bi* y *Microsoft Azure* se consigue mediante una sincronización de cuentas entre estas dos aplicaciones.

3.2. Simulación del proceso

El modelo simulado es controlado mediante una Interfaz Gráfica de Usuario (GUI) que simula la planta real. Esta desarrollada con *GUIDE*, de *Matlab*. Esta controla el inicio y paro de la simulación; las válvulas de la planta y permite hacer una parada de emergencia. En el *Anexo I* se muestra el código de programación de la Interfaz Gráfica de Usuario.

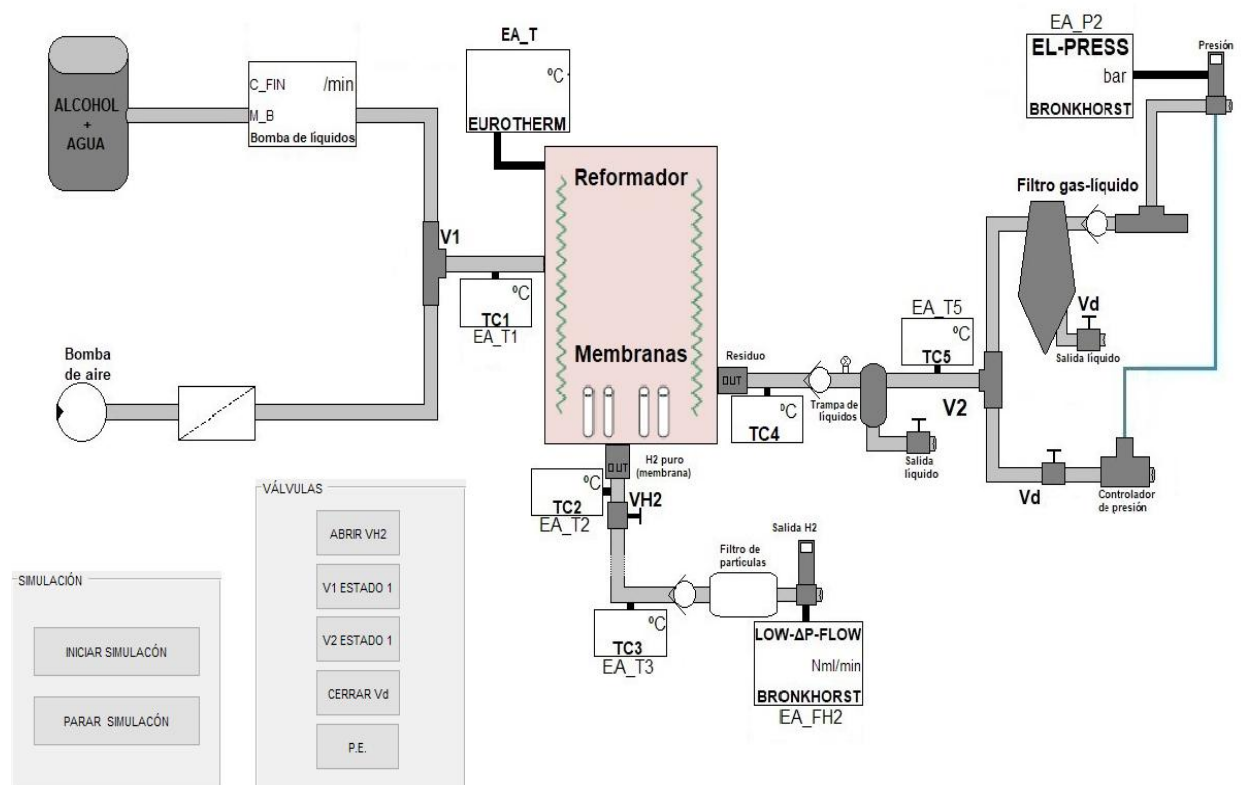


Ilustración 4: GUI de la planta.

El modelo de simulación se ha estructurado en tres bloques: datos recibidos, simulación del sistema y datos enviados.

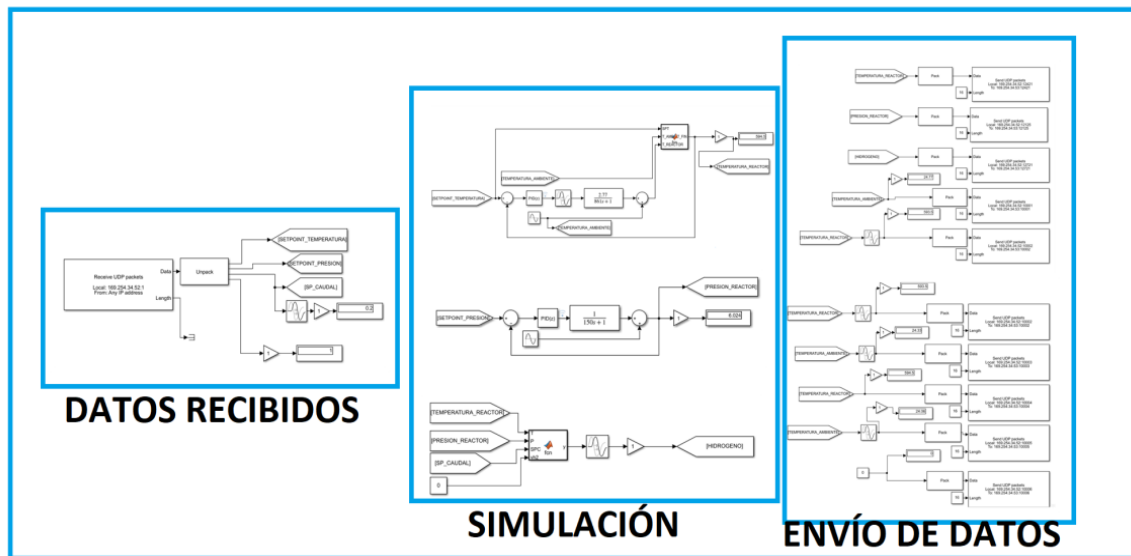


Ilustración 5: Estructura de la simulación.

A continuación, se explican detalladamente cada uno de estos tres bloques.

3.2.1. Datos recibidos.

En esta parte se recogen los datos que provienen del PLC.

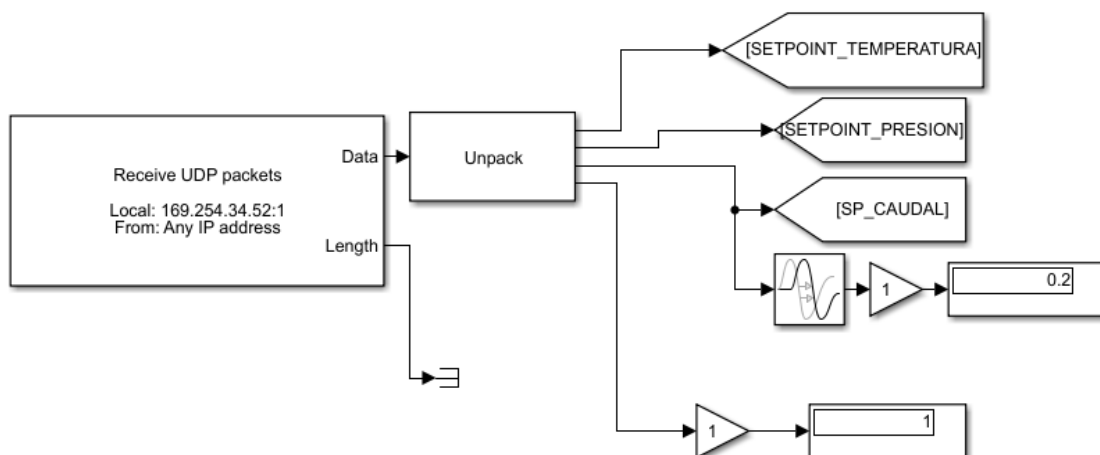


Ilustración 6: Recepción de datos en Simulink.

En la salida de datos UDP sale un vector de 32 bytes; enviado por el PLC. El dato viaja por el puerto 1 de la dirección IP "169.254.34.52". Este entra en el bloque *unpack*, que divide el vector en 4 partes iguales, formando así 4 vectores diferentes de 8 bytes cada

uno. Cada vector representa un valor (consigna de temperatura, consigna de presión, consigna de caudal de entrada y marcha de la bomba de líquidos). Este bloque pasa los 4 vectores del tipo binario a *double*.

3.2.2. Simulación

Esta es la parte donde se modela el comportamiento del sistema. Está dividido en tres partes: el modelo de la temperatura del reactor, el de la presión y el de hidrogeno. A continuación, explican cada una de estas tres:

TEMPERATURA DEL REACTOR:

Se añade la función de transferencia obtenida en el apartado 2.1.4. *Modelo matemático del proceso*, sumándole a la salida el valor de la temperatura ambiental. Esta se ha representado como un valor que oscila entre los 24 y 26 °C con una onda sinusoidal con un offset 25 y amplitud 1.

Como el controlador de temperatura de la planta real incorpora su propio PID, también se tiene que simular la acción PID del controlador. Se trata de un regulador PI, con una constante proporcional de 7 y un valor integral de 0.01. En el apartado 4. *Pruebas y resultados*, se muestra el proceso de sintonización del PID.

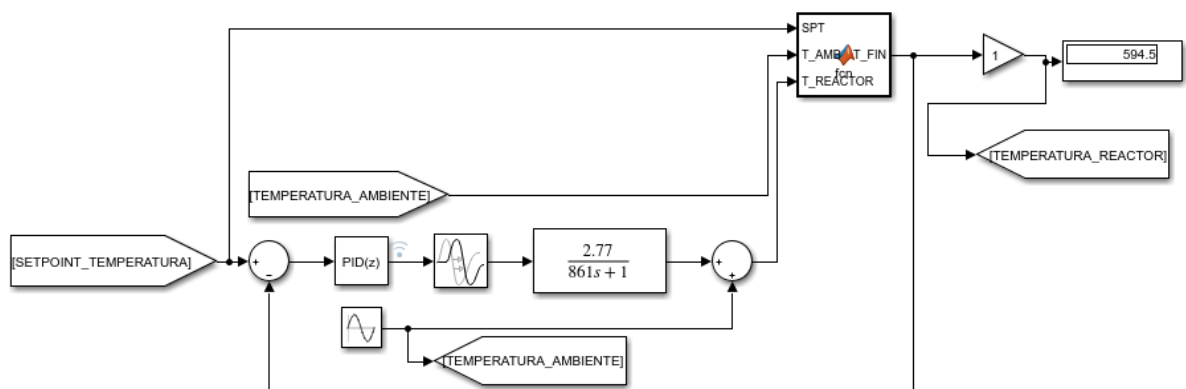


Ilustración 7: Modelo simulado de la temperatura implementado en Simulink.

PRESIÓN DEL REACTOR:

Se añade la función de transferencia obtenida en el apartado 2.1.4. *Modelo matemático del proceso*, sumándole a la salida el valor de la presión atmosférica. Esta se ha representado como un valor que oscila entre los 0.9 y 1.1 bar, con una onda sinusoidal de offset 1 y amplitud 0.1.

Al igual que con el caso anterior, el controlador de temperatura incorpora un PID, también se tiene que simular la acción PID del controlador. En este caso, se trata de un regulador PI, con un valor proporcional de 70 y un valor integral de 0.5. En el apartado 4. *Pruebas y resultados*, se muestra el proceso de sintonización del PID.

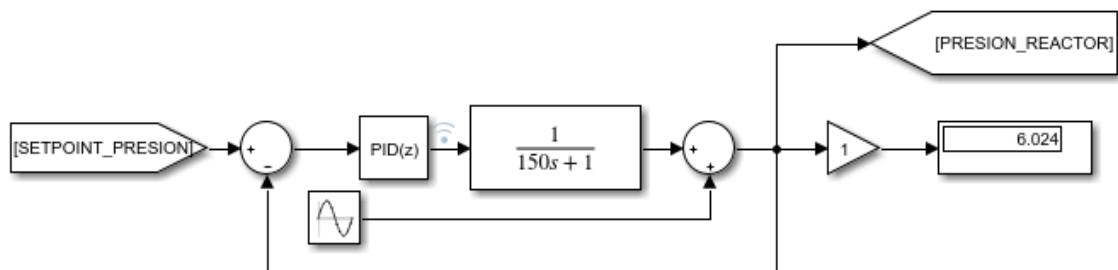


Ilustración 8: Modelo simulado de la temperatura implementado en Simulink.

SALIDA DE HIDRÓGENO:

El caudal de salida de hidrogeno varía según la temperatura y presión del reactor y el caudal de entrada en cada momento. Para lograr simular esto, se crea una función que interpole los valores de las tablas presentadas en el apartado 2.1.4. *Modelo matemático del proceso*. El bloque de funciones de la siguiente imagen es la que se encarga de ello.

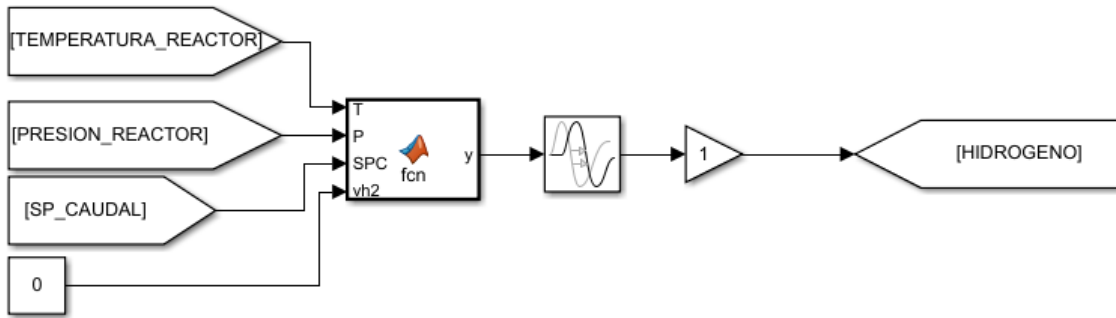


Ilustración 9: Modelo simulado de la salida de hidrógeno implementado en Simulink.

```
function y = fcn(T, P, SPC, vh2)
y = 0;
if vh2 == 1
    if SPC == 0.2
        tabla_t_p = [0,0,1,2,4,6,8,10,12;
450,0,12,23,45,66,87,99,120;
500,0,20,29.09,47.27,65.45,83.64,101.82,120;
550,0,30,38.18,54.54,70.9,87.26,103.63,120;
600,0,40,47.2727,61.8181,76.36,90.91,105.45,120]
        [a,Indice_t_min] = min(abs(tabla_t_p(2:5,1) - T))
        Indice_t_min = Indice_t_min + 1
        [b,Indice_p_min] = min(abs(tabla_t_p(1,2:8) - P))
        Indice_p_min = Indice_p_min + 1
        if tabla_t_p(1,Indice_p_min) - P > 0
            y = tabla_t_p(Indice_t_min,Indice_p_min-1) + (tabla_t_p(Indice_t_min,Indice_p_min) - tabla_t_p(Indice_t_min,Indice_p_min-1))/
            (tabla_t_p(1,Indice_p_min) - tabla_t_p(1,Indice_p_min-1))*(P-tabla_t_p(1,Indice_p_min-1));
        elseif (tabla_t_p(1,Indice_p_min) - P < 0)
            y = tabla_t_p(Indice_t_min,Indice_p_min) + (tabla_t_p(Indice_t_min,Indice_p_min+1) - tabla_t_p(Indice_t_min,Indice_p_min))/
            (tabla_t_p(1,Indice_p_min+1) - tabla_t_p(1,Indice_p_min))*(P- tabla_t_p(1,Indice_p_min));
        end
    elseif SPC == 0.02
        tabla_t_p = [0,0,1,2,4,6,8,10,12;
450,0,0.12,0.23,0.45,0.66,0.87,0.99,1.2;
500,0,.20,.29,0.47,.65,.83,1,1.20;
550,0,.30,.38,.54,.71,.87,1.03,1.20;
600,0,.40,.47,.62,.76,0.90,1.12,1.20]
        [a,Indice_t_min] = min(abs(tabla_t_p(2:5,1) - T))
        Indice_t_min = Indice_t_min + 1
        [b,Indice_p_min] = min(abs(tabla_t_p(1,2:8) - P))
        Indice_p_min = Indice_p_min + 1
        if tabla_t_p(1,Indice_p_min) - P > 0
            y = tabla_t_p(Indice_t_min,Indice_p_min-1) + (tabla_t_p(Indice_t_min,Indice_p_min) - tabla_t_p(Indice_t_min,Indice_p_min-1))/
            (tabla_t_p(1,Indice_p_min) - tabla_t_p(1,Indice_p_min-1))*(P-tabla_t_p(1,Indice_p_min-1));
        elseif (tabla_t_p(1,Indice_p_min) - P < 0)
            y = tabla_t_p(Indice_t_min,Indice_p_min) + (tabla_t_p(Indice_t_min,Indice_p_min+1) - tabla_t_p(Indice_t_min,Indice_p_min))/
            (tabla_t_p(1,Indice_p_min+1) - tabla_t_p(1,Indice_p_min))*(P- tabla_t_p(1,Indice_p_min));
        end
    else
        y = 0;
    end
elseif vh2 == 0
    y = 0;
end
```

Ilustración 10: Función dentro del bloque de funciones del Modelo simulado de la salida de hidrogeno.

3.2.3. Envío de datos.

De la misma manera que los datos recibidos vienen en un vector de 8 bytes cada dato, los datos que se envíen tienen que llegar al PLC con el mismo formato. Por tanto, se tienen que convertir los valores de tipo *double* a *uint8*. Esto se hace mediante el bloque *pack*. A diferencia de los datos recibidos que vienen todos en un mismo puerto, los datos enviados se envían uno por cada puerto. En este caso, en la dirección IP del controlador "169.254.34.53".

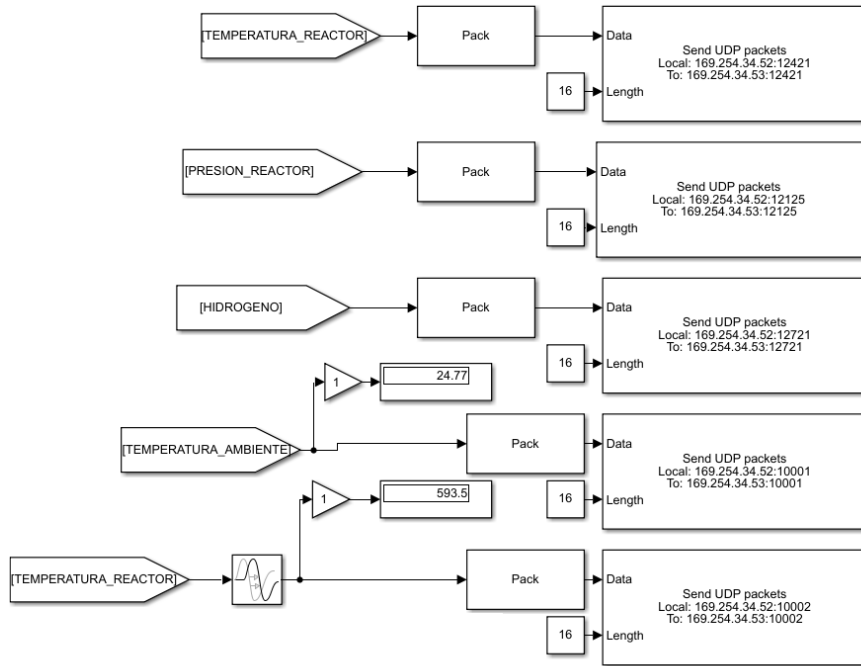


Ilustración 11: Envío de datos desde Simulink (1).

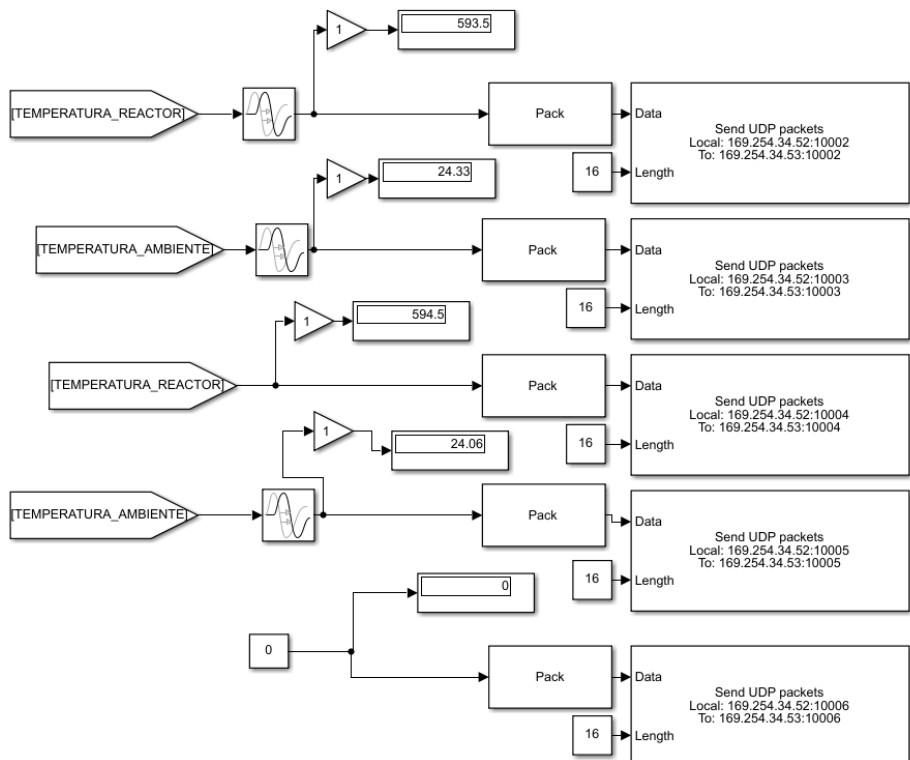


Ilustración 12: Envío de datos desde Simulink (2).

3.3. Programación del sistema de control.

Los algoritmos programados en el controlador no solo definen la lógica del control, sino que también la interfaz de usuario que permite al operario actuar sobre el sistema. Tal y como ya se ha comentado anteriormente, se utilizan dos herramientas del PLC, la aplicación de tiempo real (*Real-Time Application*) y *Node-Red*. A continuación, se comenta el programa realizado en ambas herramientas.

3.3.1. Real Time Application.

3.3.1.1. Definición de los tipos de datos.

Los datos con los que trabaja la aplicación en tiempo real se llaman variables globales. Estas no solo se pueden usar en el apartado de programación del *RTA*, sino que también pueden ser enviados al *Node-Red*.

En función de la función de cada variable, se les ha asignado de una manera determinada. A continuación, se presentan todas y cada una de ellas:

Variables del tipo IM_: Son variables booleanas cuya función es mostrar un mensaje, imagen o *display* en la interfaz de visualización (= 1 muestra, = 0 no muestra).

NOMBRE	TIPO	VALOR INICIAL	FUNCIÓN
IM_A_0	BOOL	0	Muestra mensaje
IM_A_1	BOOL	0	Muestra mensaje
IM_A_2	BOOL	0	Muestra mensaje
IM_A_3	BOOL	0	Muestra mensaje
IM_A_4	BOOL	0	Muestra mensaje
IM_PAR_1	BOOL	0	Muestra mensaje
IM_PAR_2	BOOL	0	Muestra mensaje
IM_PAR_3	BOOL	0	Muestra mensaje
IM_PAR_4	BOOL	0	Muestra mensaje
IM_PRE_0	BOOL	1	Muestra mensaje
IM_PRE_1	BOOL	0	Muestra mensaje
IM_PRE_2	BOOL	0	Muestra mensaje
IM_PRE_3	BOOL	0	Muestra mensaje
IM_F_1	BOOL	0	Muestra mensaje
IM_F_2	BOOL	0	Muestra mensaje
IM_F_3	BOOL	0	Muestra mensaje
IM_FH2	BOOL	0	Muestra flechas del caudal de H ₂
IM_FIN	BOOL	0	Muestra flechas del caudal inicial
IM_INICIO_VER	BOOL	1	Muestra botón inicio
IM_PARO_VER	BOOL	0	Muestra botón parada

IM_SP_P_ALTER NATIVO	BOOL	0	Muestra <i>display</i> de presión
----------------------	------	---	-----------------------------------

Tabla 7: Tabla donde se muestran las variables tipo IM_.

Variables del tipo IB_: Son variables booleanas cuyo valor se modifica cuando se pulsa un botón de la interfaz (1 = botón pulsado, = botón no pulsado).

NOMBRE	TIPO	VALOR INICIAL	FUNCIÓN
IB_AUTOMATICO	BOOL	0	Botón automático
IB_MANUAL	BOOL	0	Botón manual
IB_INICIO	BOOL	0	Botón inicio
IB_PARADA	BOOL	0	Botón parada
IB_START_MANU	BOOL	0	Botón start manual
IB_START_AUTO	BOOL	0	Botón start automático
IB_UP	BOOL	0	Botón subir temperatura
IB_DOWN	BOOL	0	Botón bajar temperatura
IB_OK_A_0	BOOL	0	Botón OK mensaje
IB_OK_A_1	BOOL	0	Botón OK mensaje
IB_OK_A_2_Y_3	BOOL	0	Botón OK mensaje
IB_OK_A_4	BOOL	0	Botón OK mensaje
IB_OK_PAR_1	BOOL	0	Botón OK mensaje
IB_OK_PAR_2	BOOL	0	Botón OK mensaje
IB_OK_PAR_3	BOOL	0	Botón OK mensaje
IB_OK_PAR_4	BOOL	0	Botón OK mensaje
IB_OK_PRE_0	BOOL	0	Botón OK mensaje
IB_OK_PRE_1	BOOL	0	Botón OK mensaje
IB_OK_PRE_2	BOOL	0	Botón OK mensaje
IB_OK_PRE_3	BOOL	0	Botón OK mensaje

Tabla 8: Tabla donde se muestran las variables tipo IB_.

Variables del tipo E_: Son variables booleanas que representan estados (= 1 activado, = 0 desactivado).

NOMBRE	TIPO	VALOR INICIAL	FUNCIÓN
E_AUTOMATICO	BOOL	0	Estado automático
E_MANUAL	BOOL	0	Estado manual
E_INICIO	BOOL	0	Estado inicio
E_PARO	BOOL	0	Estado paro
E_PREPARACION	BOOL	1	Estado preparación

Tabla 9: Tabla donde se muestran las variables tipo E_.

Variabes del tipo EA_: Son variables provenientes de la simulación. Son del tipo real.

NOMBRE	TIPO	VALOR INICIAL	FUNCIÓN
EA_FH2	REAL	0	Caudal de H ₂ actual
EA_P	REAL	0	Presión actual
EA_T	REAL	0	Temperatura del reactor actual
EA_T1	REAL	0	Temperatura del termopar 1
EA_T2	REAL	0	Temperatura del termopar 2
EA_T3	REAL	0	Temperatura del termopar 3
EA_T4	REAL	0	Temperatura del termopar 4
EA_T5	REAL	0	Temperatura del termopar 5

Tabla 10: Tabla donde se muestran las variables tipo EA_

Variabes del tipo C_: Son variables del tipo REAL que representan valores de consignas.

NOMBRE	TIPO	VALOR INICIAL	FUNCIÓN
C_FIN	REAL	0	Consigna caudal de entrada
C_H2	REAL	0	Consigna de caudal de H ₂
C_P	REAL	1	Consigna de presión
C_T	REAL	25	Consigna temperatura

Tabla 11: Tabla donde se muestran las variables tipo C_.

Variabes del tipo V_: Son variables del tipo REAL o BOOL utilizadas con el propósito de guardar datos temporalmente.

NOMBRE	TIPO	VALOR INICIAL	DESCRIPCIÓN
V_C_H2	REAL	0	Consigna de caudal de H ₂
V_C_P_MANU	REAL	0	Consigna de presión
V_C_P_MANU_2	REAL	0	Consigna de presión
V_C_T_MANU	REAL	450	Consigna de temperatura
V_FIN_OK	BOOL	0	Estado que determina si el caudal inicial está en marcha
V_TIN_OK	BOOL	0	Estado que determina si a temperatura inicial está en marcha
V_PE	BOOL	0	Valor que determina si se ha pulsado el paro de emergencia o no
V_SP_P_OK	BOOL	0	Valor que determina cuando se ha alcanzado SP de presión
V_SP_T_OK	BOOL	0	Valor que determina cuando se ha alcanzado SP de temperatura

Tabla 12: Tabla donde se muestran las variables tipo V_.

Variabes del tipo NR_: Son variables del tipo BOOL que viajan del RTA a Node-Red y viceversa. Funcionan como variables que indican a determinadas funciones de Node-Red cuando ejecutarse. Las salidas de estas funciones utilizan este mismo tipo de variable para indicar al RTA que el proceso de la función ha terminado y envían el resultado obtenido.

NOMBRE	TIPO	VALOR INICIAL	DESCRIPCIÓN
NR_INICIO	BOOL	0	RTA indica a <i>Node-Red</i> que se ha iniciado el proceso
NR_START_AUTO	BOOL	0	RTA indica a <i>Node-Red</i> que se ha pulsado el botón start_automático
NR_RECIBIDO	BOOL	0	RTA indica a <i>Node-Red</i> que ha recibido valor calculado

Tabla 13: Tabla donde se muestran las variables tipo NR_.

3.3.1.2. Programa

El programa creado se divide en 5 subfunciones. Estas son: funciones, preparación, arranque, funcionamiento y parada. A continuación, se explica el funcionamiento de estas.



Ilustración 13: Funciones del programa del RTA.

(En este apartado se explica que hace cada parte del programa, sin mostrarlo. Para ver el programa ir al Anexo I).

FUNCIONES:

En esta función se crean algunas subfunciones que facilitan la programación. Esto incluye:

- Una función que pone a "1" la variable $V_{SP_T_OK}$ cuando el valor de la temperatura ha llegado a la consigna.
- Una función que pone a "1" la variable $V_{SP_P_OK}$ cuando el valor de la presión ha llegado a la consigna.
- Una función que muestre en la interfaz flechas que simulan el caudal de hidrogeno y caudal de entrada.

PREPARACIÓN:

En esta parte del código se programa la secuencia de mensajes que se muestran en la interfaz que guían al operario en la parte previa del arranque de la planta. Estos mensajes indican al operario que debe hacer en cada momento con las válvulas de la planta y cuando iniciar el proceso.

ARRANQUE:

Una vez el operario ha realizado la fase de preparación de la planta, se ejecuta este bloque de programa, que permite arrancar el proceso. Esta parte consiste en establecer una temperatura inicial y un caudal de entrada.

Cuando el operario pulsa el botón *INICIO* de la interfaz, se establece un valor de consigna de la temperatura de 450 °C. A demás, una vez se pulsa el botón *INICIO*, este desaparece y aparece el botón *PARO* en su lugar.

Cuando la temperatura alcanza este valor de consigna (y antes de seleccionar alguno de los dos modos de operación) se activa la bomba de líquidos y se establece una consigna de 0.02 ml/min. A continuación, se muestra un mensaje que indica que la válvula que permite el paso del hidrogeno debe de abrirse para comprobar que el sistema está produciendo hidrogeno adecuadamente. Cuando se pulsa el botón *OK*, desaparece el mensaje y aparece otro que indica que se debe de seleccionar un modo de operación (modo manual o automático).

FUNCIONAMIENTO:

En esta parte del código se programa todo el proceso de funcionamiento normal. Es decir, desde que se selecciona un modo de funcionamiento hasta que se para el proceso.

Inicialmente, se programa la selección de modos. Cuando se selecciona por primera vez uno de los dos modos, se consigna el caudal de entrada de la bomba a 0.2 ml/min.

En cuanto al estado manual: Se programa la selección de temperatura mediante botones ascendentes y descendentes, que permiten coger valores de temperatura de 450 °C a 600 °C, en intervalos de 50, y la selección de presión de forma manual. También un mensaje de error en caso de elegir valores de presión superiores a los permitidos por el sistema.

En cuanto al estado automático: Se programa que cada vez que se pulse el estado automático, la consigna de temperatura inmediatamente se ponga a 450 °C. También la selección del valor de hidrogeno deseado y el envío del valor hacia *Node-Red*, para que desde ahí se calcule la presión correspondiente a dicho valor de hidrogeno a 450 °C. También se programa un mensaje de error en caso de elegir valores de hidrogeno superiores a los que el sistema puede producir y otro cuando se selecciona el caudal de hidrogeno de salida antes de que la temperatura haya alcanzado los 450 °C.

Se programa también que, en ambos modos de operación, sea primero la magnitud de la temperatura la que llegue a la consigna deseada y una vez esta llegue, la presión.

PARADA:

En esta parte se programa la parada del sistema.

En cuanto se pulsa el botón *PARO* en la interfaz gráfica o *P.E.* de la GUI del *Simulink*, se pone la consigna de presión a 1 bar. Cuando la presión llega a este valor aparece un mensaje para que el operario abra la válvula que permite el funcionamiento de la planta a presión atmosférica. A continuación, se detiene la bomba de líquidos y se pone la consigna de temperatura a 25 °C. Una vez la planta alcanza la temperatura

ambiente, se vuelve a empezar el proceso desde la función de *PREPARACIÓN*, para iniciar el proceso de nuevo en cualquier momento.

3.3.2. Node-red.

En este apartado se explica el programa del *Node-Red*. Este se divide en varios *flows*, con una función distinta cada uno. Estos son: Envío de datos a Simulink, recepción de datos de Simulink, guardado de datos, cálculo de presión en modo automático y envío de datos al Centro IoT. De este último se habla en el siguiente apartado.

ENVÍO DE DATOS:

Inicialmente, se recogen los valores de C_T, C_P, C_FIN y M_B de la RTA que se actualizan cada segundo. Se hace una conversión de estos valores tipo *float* a *buffer*, donde se empaqueta el valor en un vector de 8 bytes. Posteriormente, se almacenan los 4 valores distintos en un mismo *buffer*, obteniendo por tanto un vector de 32 bytes. Se envía el valor al puerto 1 de la dirección IP “169.254.34.52”.

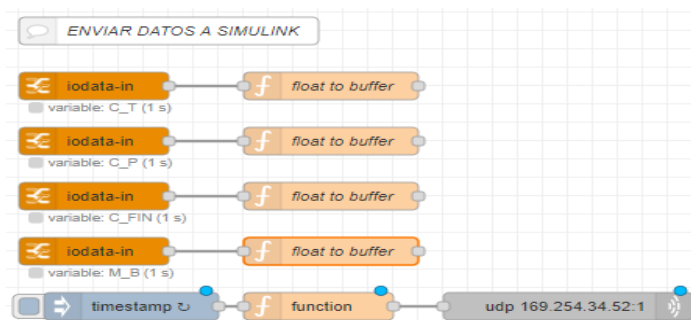


Ilustración 14: Envío de datos a Simulink.

RECEPCIÓN DE DATOS:

A diferencia del envío de datos, los datos recibidos entran cada uno por un puerto distinto. El dato entrante se pasa de *buffer* a *float*, y se pasa el valor a la RTA, cada dato a su variable global correspondiente. Los valores de EA_T, EA_P, EA_C y EA_FH2 se guardan globalmente en un espacio de memoria del Node-Red puesto que posteriormente serán necesarios.

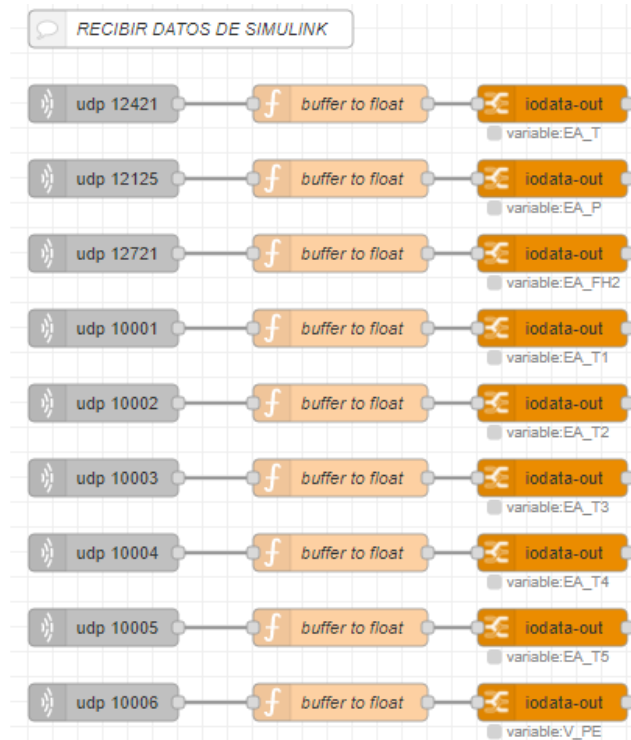


Ilustración 15: Recepción de datos de Simulink.

GUARDADO DE DATOS Y CREACIÓN DE INTERFAZ PARA VISUALIZARLOS.

Esta parte del programa crea archivos donde se guardan las variables medidas por el sistema. También desarrolla la interfaz que permite graficar, descargar o eliminar estos archivos. Para desarrollar esta parte es necesario la instalación de los nodos “node-red-contrib-fs” y “node-red-dashboard”.

Concretamente, cada vez que se inicie un nuevo proceso (se pulse el botón iniciar) se crea un archivo de tipo “.csv” con nombre “archivo_iniciado_en_dd_mm_aaaa_a_las_hh_mm.csv”. En este archivo se guardan cada cinco segundos los valores de la temperatura, presión e hidrogeno. Estos datos se guardan en la tarjeta MicroSD de 32 GB. Una vez se para el proceso, se deja de guardar.

Selector de archivos

ACTUALIZAR /run/media/mmcbk0p1/archivo_iniciado_en_8_4_2021_a_las_14_5.csv BORRAR GRÁFICO

File Name	Size	Created	Changed
archivo iniciado en 19 4 2021 a las 25 7.csv	4311	2021-04-19 23:14:34	2021-04-19 23:14:34
archivo iniciado en 20 4 2021 a las 2 23.csv	744	2021-04-20 00:23:59	2021-04-20 00:23:58
archivo iniciado en 8 4 2021 a las 14 5.csv	2968	2021-04-08 12:11:05	2021-04-08 12:11:04
archivo iniciado en 8 4 2021 a las 16 17.csv	1027	2021-04-08 14:24:54	2021-04-08 14:24:54
archivo iniciado en 8 4 2021 a las 16 24.csv	3980	2021-04-08 14:33:42	2021-04-08 14:33:42
archivo iniciado en 8 4 2021 a las 16 33.csv	2757	2021-04-08 14:37:28	2021-04-08 14:37:28
archivo iniciado en 8 4 2021 a las 17 20.csv	6001	2021-04-08 15:28:50	2021-04-08 15:28:50
archivo iniciado en 8 4 2021 a las 19 45.csv	118189	2021-04-08 20:29:02	2021-04-08 20:29:02

Ilustración 16: Interfaz desde la cual se puede graficar, eliminar o descargar los archivos creados.

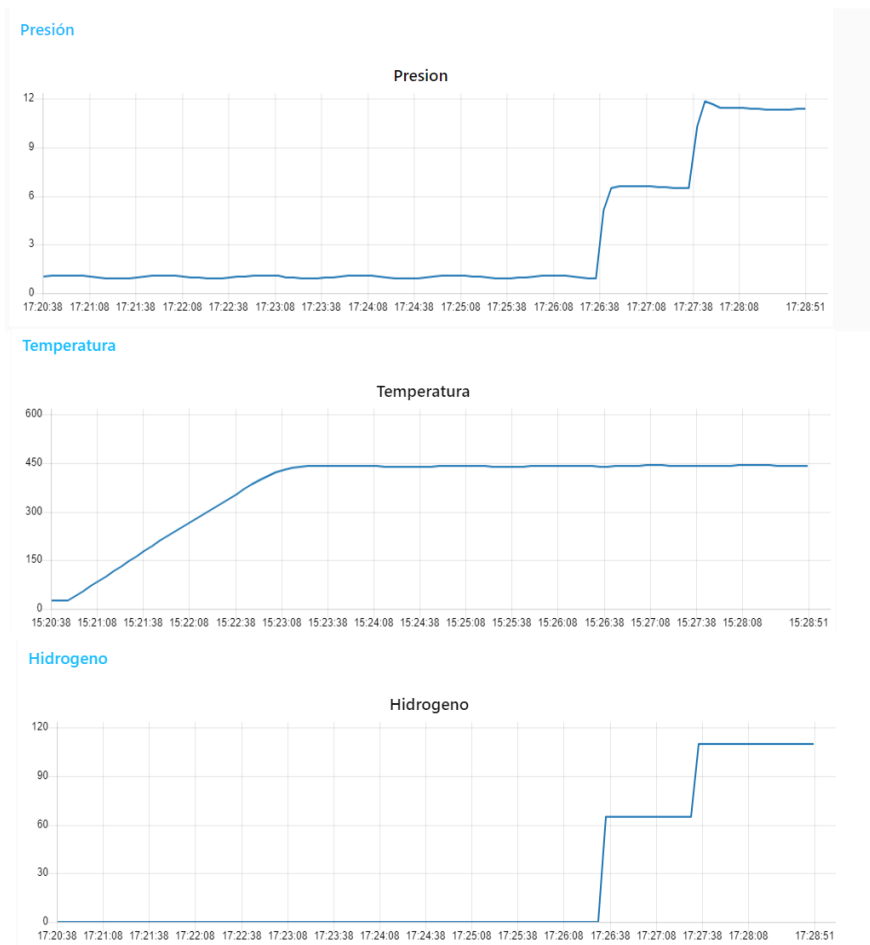


Ilustración 17: Visualización de los datos guardados en gráficos lineales.

CALCULO DE PRESIÓN EN MODO AUTOMÁTICO:

En este apartado se crea una función que recibe el valor de consigna de hidrogeno deseado en el estado automático y calcula el valor de presión que le corresponde esa salida de hidrogeno a 450 °C.

Cuando en la interfaz de control se selecciona el valor de hidrogeno deseado y se pulsa el botón aceptar, la variable `NR_START_AUTO` se pone a "1". Solo cuando esta variable es "1", la función `obtener presión` calcula el valor de presión y lo envía a la variable global `C_P` de la `RTA`. A continuación, el valor `NR_RECIBIDO` se pone a "1", indicando a la `RTA` que el valor ya ha sido calculado, para que desde el `RTA` la variable `NR_START_AUTO` se vuelva a poner a "0".

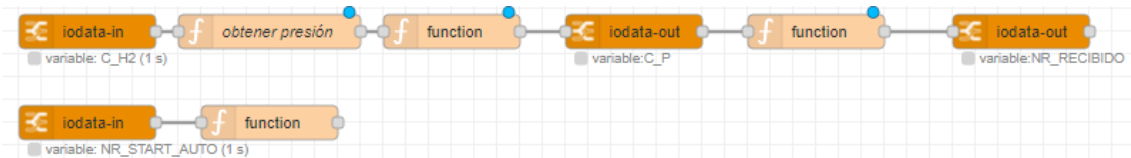


Ilustración 18: Flow de Node-Red que calcula la presión en el estado automático.

3.4. Creación del panel IoT.

Este es el apartado donde finalmente se explica la solución al objetivo principal. La solución IoT consiste en un panel que monitoriza la planta en tiempo real.

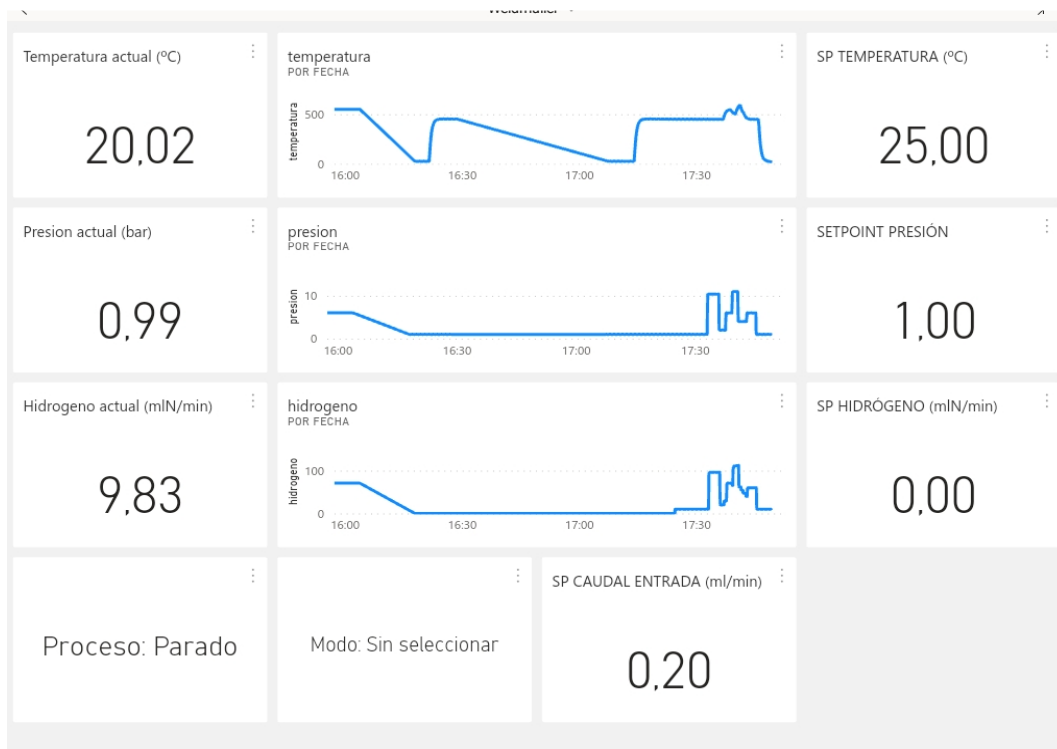


Ilustración 19: Panel IoT.

El panel muestra tres gráficas lineales de la evolución de las variables de temperatura, presión y producción de hidrogeno respecto del tiempo. En el lado izquierdo de cada

gráfica hay una tarjeta que muestra el valor dicha variable en ese momento. En el lado derecho de la gráfica se muestra el valor de consigna de cada variable. Finalmente, en la parte inferior se observa el estado del proceso (en marcha o parado), el modo de operación (automático, manual o no seleccionado) y el valor de consigna de la bomba de líquidos.

Las herramientas utilizadas para esta solución son *Azure* y *Power BI*. La comunicación que se establece entre los programas y el PLC queda resumida con la siguiente imagen:

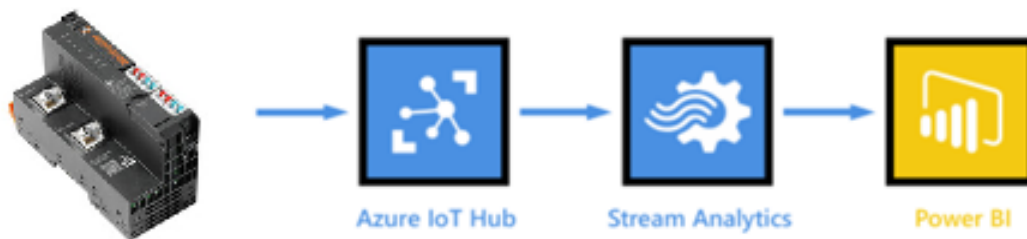


Ilustración 20: Comunicación entre las aplicaciones utilizadas para la solución IoT.

Azure IoT Hub o *Centro de IoT* de *Azure* es el que actúa como intermediario entre nuestro PLC y la aplicación IoT final. El *Stream Analytics* procesa los datos que recibe el *Centro de IoT* y los envía a *Power BI*, donde se crea el panel de visualización.

Lo primero que se tiene que hacer es crear el Centro de IoT. Para esto se introduce un nombre, la región en la que se quiere situar (en nuestro caso, oeste de Europa), el tipo de suscripción disponible y el grupo de recursos (en el mismo momento en el que se crea el centro de IoT se puede crear muy fácil y rápidamente un grupo de recursos). Debido a que la suscripción de la que se dispone es la gratuita, el centro de IoT está limitado a 8000 mensajes al día. Esto nos limita el tiempo de actualización de los datos a 10 segundos, si se tuviera el sistema en marcha durante todo el día. Aun así, para la solución que se quiere implementar, es un buen tiempo de actualización.

Para terminar de configurar nuestro Centro de IoT, se crea un nuevo dispositivo IoT llamado "UControl".

Id. de dispositivo	Estado	Última actualización de...	Tipo de autenticación	Recuento de mensajes de la nube al dispositivo
UControl	Enabled	--	Sas	0

Ilustración 21: Creación del nuevo dispositivo.

3.4.1. Envío de datos a Azure.

Una vez ya se ha configurado la Centro de IoT, es hora de enviarle los datos desde nuestro PLC. Esto se hace desde el *Node-Red*.

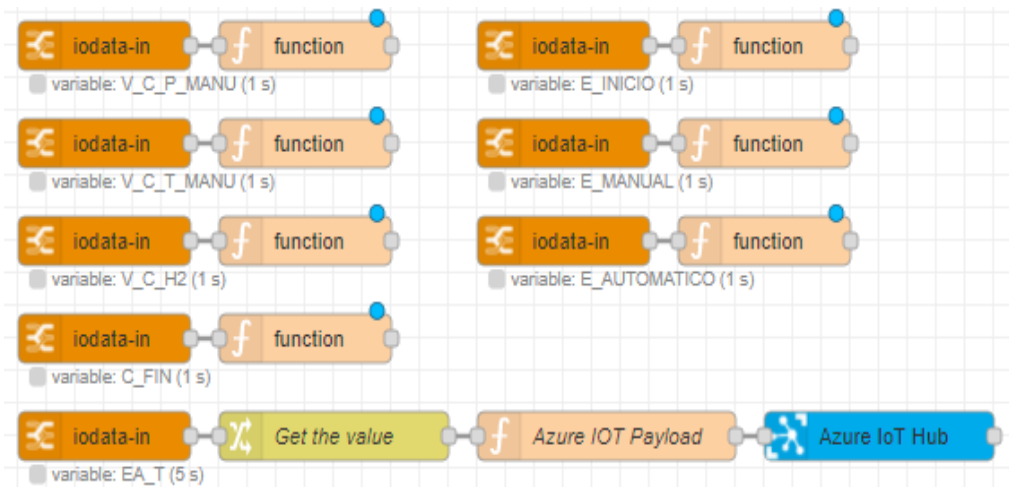


Ilustración 22: Envío de datos a Microsoft Azure.

Inicialmente, se traen las variables que se quieren mostrar en el panel, de la *RTA* al *Node-Red*. A continuación, se introduce al nodo *Azure IoT Hub* el nombre del Host del Centro IoT y se envían los datos. Los mensajes enviados al Centro IoT deben de enviarse con una estructura determinada, donde se indique el nombre del dispositivo creado al que se quiere enviar, su clave primaria, el protocolo de comunicación y, finalmente, los datos que se quieren enviar. Se ha configurado el *Node-Red* para que envíe este mensaje cada 5 segundos.

Consulta de prueba Guardar consulta Descartar cambios

```

1 SELECT
2     temperatura as TEMPERATURA,
3     presion as PRESION,
4     hidrogeno as HIDROGENO,
5     IoTHub.EnqueueedTime as FECHA,
6     auto AS ESTADO_AUTOMATICO,
7     manu as ESTADO_MANUAL,
8     time as time_stamp,
9     SP_T_M as SETPOINT_T_M,
10    SP_P_M AS SETPOINT_P_M,
11    SP_H2_A AS SETPOINT_H2_A,
12    SP_FIN as SETPOINT_F_IN,
13    E_INICIO AS E_INICIO
14
15 INTO
16     SalidaPBI
17
18 FROM
19     EntradaDeDatos

```

Ilustración 24: Selección de entradas y salidas de los datos y de los datos que se desean utilizar a Power BI.

Antes de iniciar el trabajo de *Stream Analytics*, se pueden hacer consultas de las entradas y las salidas:

temperat...	presion	hidrogeno	auto	manu	SP_T_M	SP_P_M	SP_H2_A	SP_FIN	E_INICIO	time	EventProcessedUtcT...
25.97	1	0	0	0	25	1	0	0	0	1619163271064	"2021-04-23T07:34:42...
25.96	1.01	0	0	0	25	1	0	0	0	1619163266068	"2021-04-23T07:34:42...
25	1	0	0	0	25	1	0	0	0	1619163254819	"2021-04-23T07:34:42...

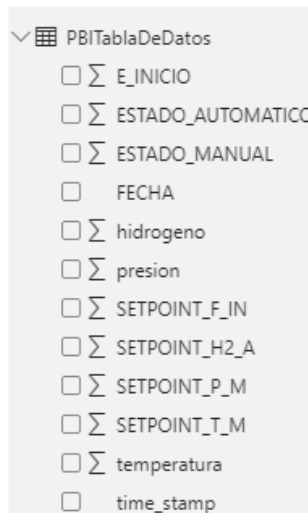
Ilustración 25: Consulta de los datos de entrada.

TEMPERATURA	PRESION	HIDROGENO	FECHA	ESTADO...	ESTADO_MANUAL	time_stamp	SETPOINT_T_M	SETPOINT_P_M	SETPOINT_H2_A	SETPOINT_F_IN	E_INICIO
25.97	1	0	"2021-04-23T07:34:31.1180000Z"	0	0	1619163271064	25	1	0	0	0
25.96	1.01	0	"2021-04-23T07:34:26.1130000Z"	0	0	1619163266068	25	1	0	0	0
25	1	0	"2021-04-23T07:34:22.9570000Z"	0	0	1619163254819	25	1	0	0	0

Ilustración 26: Consulta de los datos de salida.

En este punto, ya se ha configurado todo lo relacionado con *Azure*. A continuación, se pone en marcha el trabajo de *Stream Analytics* para que el proceso se ponga en marcha y se cree el conjunto de datos de *Power BI*.

En *Power BI Desktop*, se abre el conjunto de datos que se ha creado desde *Azure* y se observa que, efectivamente, aparecen todos datos enviados a la salida de *Power BI* desde el trabajo de *Stream Analytics* de *Azure*.



Il·lustració 27: Taula de dades de Power BI.

Power BI almacena los datos en forma de tabla, creando una columna para cada variable. En nuestro caso, cada cinco segundos aparece una nueva fila con un nuevo valor en todas las columnas de la tabla. Con estas columnas la aplicación permite la creación de gráficos lineales muy fácilmente. Por ejemplo, para graficar la evolución de la temperatura, basta con seleccionar el gráfico de líneas y poner la columna *temperatura* en el eje Y, y la columna *fecha* en el eje X. Se procede de la misma forma para graficar la evolución de la presión y del hidrogeno producido.

En cambio, para mostrar en una etiqueta el valor en tiempo real es algo un poco más complejo, puesto que hay que seleccionar el último valor recibido de la tabla. Para esto se crea una medida que calcule el valor de la columna (de temperatura, presión o hidrogeno) cuando el valor de “time_stamp” sea máximo.

```
Temperatura actual = CALCULATE(MAX(PBITablaDeDatos[temperatura]),FILTER(PBITablaDeDatos,
PBITablaDeDatos[time_stamp] = max(PBITablaDeDatos[time_stamp])))
```

Il·lustració 28: Medida que calcula la temperatura actual.

De la misma manera, se crea una medida llamada Presion_actual e Hidrogeno_Actual.

Para las tarjetas que indican estados (proceso y modo) hay que hacer algo similar, con alguna diferencia. A continuación, se muestra la medida creada para la tarjeta en la que se visualiza el modo de operación:

```

1 MODO DE OPERACIÓN =
2 var estado_manual = value(CALCULATE(MAX(PBITablaDeDatos[ESTADO_MANUAL]), FILTER
(PBITablaDeDatos,PBITablaDeDatos[time_stamp] = MAX(PBITablaDeDatos[time_stamp]))))
3 var estado_automatico = value(CALCULATE(MAX(PBITablaDeDatos[ESTADO_AUTOMATICO]), FILTER
(PBITablaDeDatos, PBITablaDeDatos[time_stamp] = MAX(PBITablaDeDatos[time_stamp]))))
4 return if(estado_automatico = 0, if(estado_manual = 1, "Modo: manual", "Modo: Sin
seleccionar"), if(estado_manual = 0, "Modo: automático"))

```

Ilustración 29: Medida que indica el modo de operación del sistema.

La línea dos del código crea una variable donde se almacena el valor de la fila mas reciente de la columna "ESTADO_MANUAL". La tercera línea hace lo mismo, pero con la columna "ESTADO_AUTOMATICO". Finalmente, se crea un condicional que devuelve un texto en función de los estados de las variables creadas.

Para la tarjeta que muestra si el proceso está en marcha o parado, se crea otra medida similar a la anterior.

```

1 PROCESO =
2 var ultimo_timestamp = MAX(PBITablaDeDatos[time_stamp])
3 var E_ESTADO_INICIO = value(CALCULATE(MAX(PBITablaDeDatos[E_INICIO]), FILTER
(PBITablaDeDatos,PBITablaDeDatos[time_stamp] = ultimo_timestamp)))
4
5 return if(E_ESTADO_INICIO = 1,"Proceso: En marcha", "Proceso: Parado")

```

Ilustración 30: Medida que indica el estado del proceso del sistema.

4. Pruebas y resultados

4.1. Comunicación entre el PLC y el Simulink.

Objeto: Comprobar que se puede establecer una comunicación entre el PLC y la planta simulada en Simulink.

Descripción: Se han planteado varios protocolos de comunicación entre el PLC y el Simulink. Finalmente, se ha decidido utilizar el protocolo UDP por ser sencillo y funcional. Se han hecho pruebas de envío y recepción de datos individuales.

Resultado: Se ha logrado establecer una comunicación en tiempo real entre Simulink y el PLC, con un periodo de muestreo lo suficientemente bueno.

4.2. Envío de datos al centro IoT de Azure.

Objeto: Comprobar que el Centro IoT de Azure creado recibe los datos enviados desde la pasarela IoT.

Descripción: Se envían al Centro IoT de Azure las variables del PLC que se quieren mostrar en el panel de IoT. Se utiliza la opción de “Consulta” que dispone la aplicación de *Stream Analytics* para comprobar si los datos llegan de forma correcta.

Resultado: Se ha logrado enviar datos al centro IoT de Azure.

temperat...	presion	hidrogeno	auto	manu	SP_T_M	SP_P_M	SP_H2_A	SP_FIN	E_INICIO	time	EventProcessedUtc...
25.97	1	0	0	0	25	1	0	0	0	1619163271064	"2021-04-23T07:34:42...
25.96	1.01	0	0	0	25	1	0	0	0	1619163266068	"2021-04-23T07:34:42...
25	1	0	0	0	25	1	0	0	0	1619163254819	"2021-04-23T07:34:42...

Ilustración 31: Muestra de valores de entrada al Centro IoT de Azure.

4.3. Comunicación entre la RTA y el Node-Red

Objeto: Comprobar que las variables globales creadas en la RTA pueden utilizarse en el *Node-Red*.

Descripción: Se crean variables de prueba en la RTA para a las que se les modifican su valor tanto desde la RTA como desde el *Node-Red*, con el objetivo de comprobar que ambas herramientas perciben los cambios realizados desde la otra plataforma.

Resultado: Se comprueba que se puede establecer comunicación en tiempo real entre la RTA y el Node-Red con un tiempo de respuesta mínimo de 50 ms.

4.4. Sintonización de los PIDs.

Objeto: Sintonizar los PIDs de los controladores de temperatura y presión simulados.

Descripción: En ambos casos se ha utilizado un método de prueba y error. Se ha procedido de la siguiente manera:

- Se incrementa la constante proporcional hasta obtener una respuesta cercana a la deseada, con un error estacionario diferente de 0.
- Se incrementa la constante integral hasta eliminar el error en estado estacionario.

Resultado: Se ha obtenido una buena respuesta para ambos controladores.

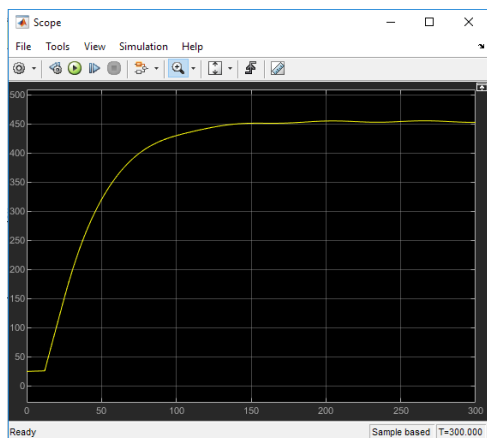


Ilustración 32: Respuesta de la función de transferencia de temperatura

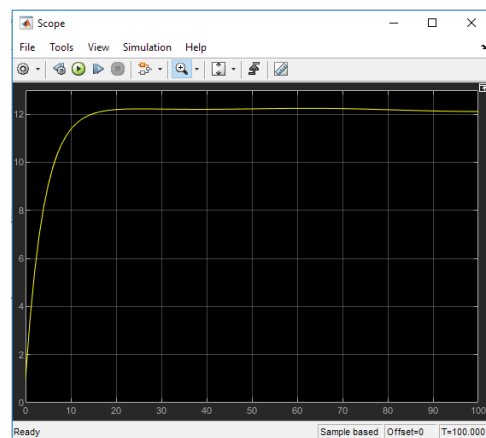


Ilustración 33: Respuesta de la función de transferencia de la presión.

5. Conclusiones.

A pesar de no disponer del modelo matemático de la planta real, se ha logrado crear una simulación totalmente práctica para poder implementar la solución propuesta como objetivo principal de este proyecto. Además, se ha creado una Interfaz Gráfica de Usuario que permite controlar esta simulación, ver las variables de la planta y modificar los estados de las válvulas, como si se tratase de la planta real.

También se ha logrado crear un sistema de control capaz de controlar las variables del sistema. Además, un sistema de control que dispone de una interfaz muy fácil de usar que permite al operario la selección de dos modos de funcionamiento: automático o manual. Esta interfaz también permite monitorizar la planta, mostrando las variables del sistema en gráficos lineales y en un dibujo de la planta. Asimismo, se ha creado un sistema que permite la visualización de las variables de la planta de días atrás, gracias al guardado de los datos en una tarjeta microSD introducida en el PLC.

Finalmente, se ha logrado cumplir el objetivo principal de este proyecto de implementar una plataforma IoT para la planta. Se ha desarrollado un panel de visualización mediante IoT que permite monitorizar las variables de la planta en tiempo real. Además, de que el panel puede verse en cualquier smartphone, tableta u ordenador por varios usuarios al mismo tiempo. Se ha conseguido que este tenga un tiempo de actualización de 5 segundos, lo que lo hace bastante rápido.

Como conclusión final, estoy satisfecho con el trabajo realizado y los resultados obtenidos.

6. Futuras líneas de trabajo.

El siguiente paso sería implementar la solución a la que se ha llegado en este proyecto, a la planta real del laboratorio.

7. Webgrafía.

- Análisis de alternativas tecnológicas para la separación de hidrogeno en procesos de decarbonizacion de metano:
http://oa.upm.es/46076/1/TFG_RAFAEL_BELLERA_FERNANDEZ_DE_LA_CRUZ_a.pdf
- H2 como fuente de energía renovables: producción por métodos catalíticos mediante el reformado de metanol:
<http://www.inin.gob.mx/documentos/publicaciones/contridelinin/Cap%C3%ADtulo%2015.pdf>
- U-control manual:
<https://download.weidmueller.com/asset/download/file//77934>
- Diseño de un sistema de control de temperatura:
<http://e-spacio.uned.es/fez/eserv/taee:congreso-2006-1116/S3F04.pdf>
- Centro de ayuda de Matlab:
<https://es.mathworks.com/help/>
- Comunidad de Power BI:
<https://community.powerbi.com/>
- Node-Red documentación:
<https://nodered.org/docs/>
- Creación de un Centro IoT en Azure:
<https://docs.microsoft.com/es-es/azure/iot-hub/iot-hub-create-through-portal>
- Visualización de datos de sensores en tiempo real desde Azure IoT Hub con Power BI:
<https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-live-data-visualization-in-power-bi>
- Los sistema de primer orden y los controladores PID:
https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiBi5OmiqfwAhVJB2MBHQaVAz4QFjAAegQIBBAD&url=http%3A%2F%2Fdea.unsj.edu.ar%2Fcontrol%2FControladoresPID.pdf&usg=AOvVaw0sQTWBwOxr-P_6JzA8yzgl

- Foro de resolución de dudas de JavaScript:
<https://es.stackoverflow.com/?tags=javascript>
- AMQP: conoce el Advanced Message Queuing Protocol:
<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/advanced-message-queuing-protocol-amqp/>
- UDP: ¿qué es el protocolo UDP?:
<https://www.ionos.es/digitalguide/servidores/know-how/udp-user-datagram-protocol/>

Anexo I: Programación.



Anexo II: Equipos y tarjetas.



Anexo III: Memoria econ3mica.

