

## 1. PLANTEAMIENTO DEL PROYECTO

El proyecto de la asignatura consiste en **desarrollar una red industrial que emule el funcionamiento de un vehículo**. En la figura 1 se muestra el esquema general de la red.

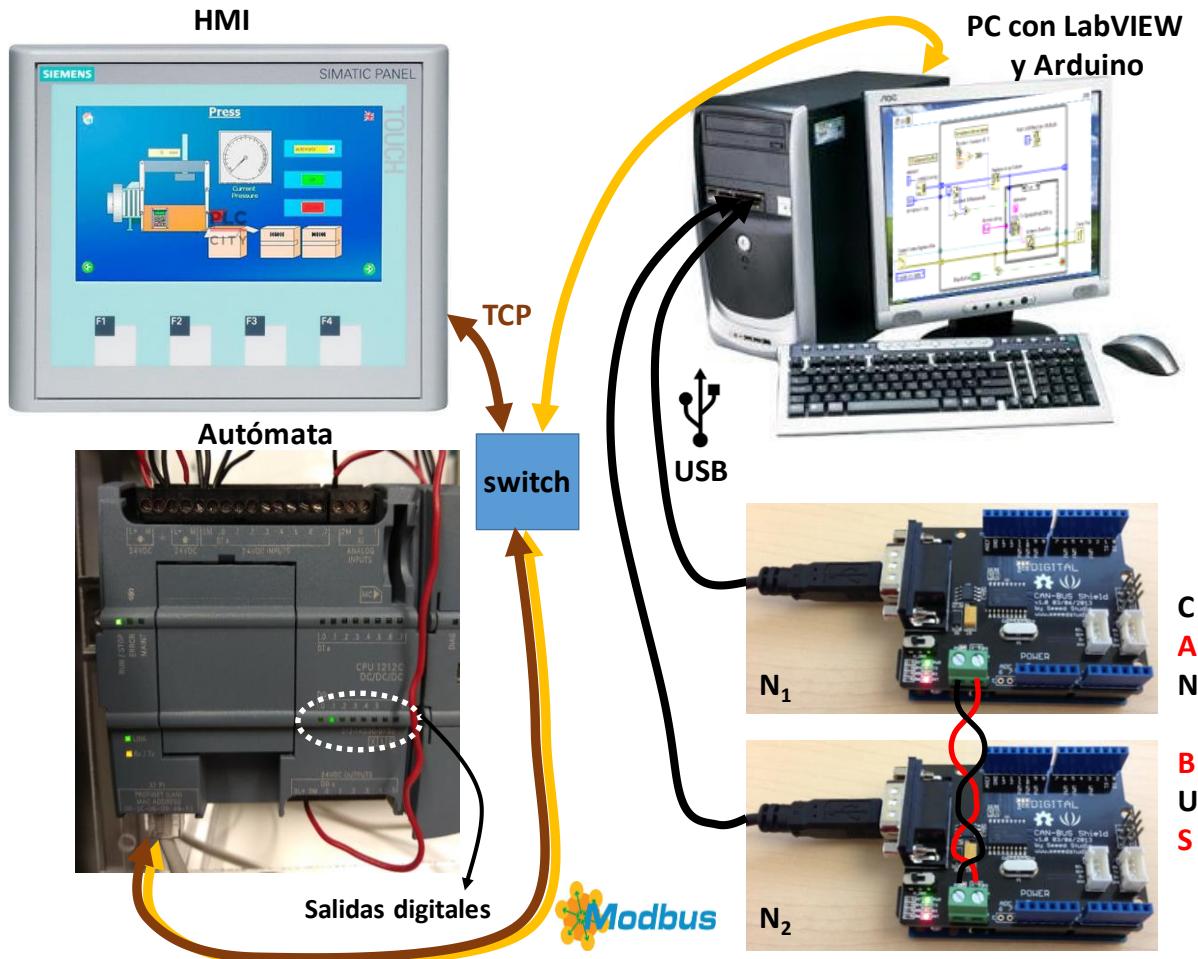


Figura 1: Esquema general del proyecto planteado.

Se pueden distinguir hasta 4 tipos de comunicación:

(a) **CAN BUS (3 ptos.):**

La red CAN bus constará de 2 nodos Arduino que se podrán programar con el IDE de Arduino. Ambos nodos irán conectados al PC. El primero de ellos (N1) interrogará al otro nodo CAN bus (N2), el cual actuará como una ECU (*Electronic Control Unit*) siguiendo el protocolo OBDII (ver apuntes del tema 5). La idea es emular la comunicación entre un dispositivo de diagnóstico de vehículos (N1) y dos de las ECUs de un vehículo (N2).

El procedimiento consistirá en enviar continuamente solicitudes de datos desde N1 a N2, a las que N2 (ECU) responderá con la trama de respuesta de datos. Se solicitarán 2 variables: una de 8 bits y otra de 16 bits. La lista de PIDs está disponible en la web: [https://en.wikipedia.org/wiki/OBD-II\\_PIDs](https://en.wikipedia.org/wiki/OBD-II_PIDs)). Cada pareja de estudiantes trabajará con unos valores de PID diferentes. La distribución de las variables y su correspondencia con la pareja que realizará el proyecto se refleja en la siguiente tabla:

Nº pareja	PID variable 2 Bytes (decimal)	PID variable 1 Byte (decimal)
1	3	17
2	12	47
3	16	4
4	20	5
5	31	6
6	33	10
7	34	14
8	35	15
9	3	47
10	12	5
11	16	6
12	20	10

Para cada variable se tendrán disponibles entre 5 y 10 datos inventados por el alumnado que simularán los datos reales de los cuales no se dispone. A modo de ejemplo, para la temperatura se puede ir enviando, progresivamente: 23,24,25,26,23,24,25,26, así una y otra vez. **Los valores que finalmente se vayan a mostrar serán valores acordes con la realidad.**

#### (b) USB (0,5 ptos.):

Los nodos Arduino se programarán desde el PC usando sus IDEs, los cuales intercambiarán datos a través del protocolo USB. Asimismo, se utilizará LabVIEW® para graficar los datos que irá enviando N1 conforme los vaya recibiendo de N2 en tiempo real.

Para ello habrá que programar ambos nodos, haciendo que se comuniquen entre ellos mediante CAN BUS y que el nodo N1 se comunique con el PC mediante USB. De cara a facilitar la programación de la aplicación de LabVIEW®, se entrega al alumnado el programa “monitorserie\_labview.vi”, para el que sólo hace falta configurar los parámetros de comunicación USB y ejecutarla. Además, dicho programa será convenientemente modificado para distinguir los diferentes parámetros que recibe del nodo N1 mediante CAN BUS y, tras ello, enviarlos al PLC S7-1200 mediante MODBUS TCP.

#### (c) MODBUS (3 ptos.):

Como se ha comentado, el valor de las variables que suministra el nodo N2 (la ECU del vehículo) al nodo N1, y que se reciben vía USB en el PC, se enviarán a su vez mediante MODBUS al PLC. Estos datos se irán mostrando en código binario en las salidas digitales del PLC.

De acuerdo con lo anterior, al PLC le llegará la orden de escribir en las salidas de la CPU sucesivamente el valor en binario de cada una de esas variables. Por ejemplo, si el dato de temperatura en ese momento es 25 grados, se deberá poner en las salidas 011001. En el caso de variables de 16 bits habrá que escribir primero el byte más significativo y luego el menos significativo.

Una vez se han escrito todas las variables se vuelve a comenzar con la primera. El lapso de tiempo en que se muestra una y otra variable lo puede elegir el alumnado. Se recomienda que sea un lapso suficiente para poder distinguir visualmente entre valores de una u otra variable, así como para que todo el sistema funcione correctamente.

El módulo I/O Server de MODBUS creado en TIA Portal sólo se usará para comprobar si la comunicación entre PC y PLC funciona y para ver cómo son los mensajes MODBUS, pero todo esto solo servirá como ayuda. En el programa de LabVIEW® será necesario emplear la paleta de bloques de

TCP vista en la práctica 2. Es decir, en algún momento será necesario rellenar el campo de dato TCP de forma adecuada para poder comunicar el PC con el PLC adecuadamente.

**(d) TCP (1,5 ptos.):**

Finalmente, además de la comunicación MODBUS TCP realizada con anterioridad, habrá que programar la HMI para que vaya mostrando los valores de las variables originalmente solicitadas a las ECUs del vehículo (datos proporcionados por el nodo N2 de Arduino). Para ello se empleará una comunicación TCP entre el PLC y la HMI, de acuerdo a lo visto en la práctica 3. De este modo, la HMI hará las funciones del “ordenador de a bordo” del coche, donde se podrá ver con iconos la evolución de las variables que se manejen.

De acuerdo con la comunicación esquematizada en la figura 1, la comunicación TCP y la MODBUS se realizan en paralelo a través del cable que conecta el PLC con el switch. Es interesante tener esto en cuenta, ya que en el caso del cable USB no se puede trabajar con LabVIEW® y Arduino al mismo tiempo. Se recuerda que el puerto serie es de uso exclusivo para una sola conexión, mientras que el cable ethernet es multiconexión.

**(e) Alarmas (2 ptos.)**

Para las dos variables que se manejen, habrá que mostrar alarmas en la HMI, en el programa de LabVIEW® y, por último, en el nodo N1 de Arduino. Todo ello, en respuesta a que se sobrepase algún límite dentro de los valores de las variables que se hayan programado.

## **2. ENTREGA DEL PROYECTO**

Se entregarán los ficheros de programación necesarios para la correcta ejecución del proyecto, así como una memoria explicativa en pdf de 10 páginas. Se sugieren los siguientes apartados para la memoria, aunque se puede seguir un esquema diferente si se prefiere:

Introducción

Objetivos

Metodología empleada: aquí se describirán los elementos hardware y software empleados y se mostrará un esquema general del sistema de comunicación

Resultados: se describirán los programas realizados, los resultados obtenidos y las dificultades que se han superado

Conclusiones

Bibliografía y referencias

## **3. METODOLOGÍA DE TRABAJO**

El tiempo estimado para la realización del proyecto es de 16 horas lectivas. Como guía, se propone seguir el siguiente orden en la consecución de objetivos:

**Sesiones de CAN BUS (2-3h):**

Aprender el manejo de la comunicación entre dos nodos CAN bus mediante el documento “Sesión CAN Bus”. Al finalizar, **se preparará un entregable para su evaluación como 10% de prácticas, junto con el resto de informes.**

### **Siguientes sesiones (13-14h):**

Tiempos estimados para cada una de las tareas a realizar:

- Comunicar los dos nodos CAN Bus entre sí intercambiándose un dato (3h).
- Comunicar las dos variables entre los nodos CAN Bus (2h).
- **Entrega del ejercicio de CAN BUS usando el laboratorio Remoto** (la fecha límite se fijará en clase)
- Visualizar los datos del nodo central de CAN Bus en el PC mediante LabVIEW® (0,5h).
- Enviar los datos del PC al autómata mediante MODBUS. Se adjunta un documento donde se indica cómo habilitar el PLC como servidor MODBUS: “PLC servidor MODBUS” (3h).
- Programar la HMI para que los valores de las variables se reflejen en ella (2h).
- Implementar el sistema de alarmas (3h).

Se recomienda también el siguiente manual de Arduino como apoyo para la programación:

<https://arduinobot.pbworks.com/f/Manual+Programacion+Arduino.pdf>