



ANÁLISIS Y ELECCIÓN DE DIFERENTES SISTEMAS PARA LA OBTENCIÓN DE DATOS DE LA CENTRALITA DE UN VEHÍCULO PARA SU POSTERIOR MUESTRA POR PANTALLA

Trabajo Fin de Grado



Febrero de 2018

Universitat Politècnica de València
Grado en Ingeniería Eléctrica

Autor: Carlos Vicente López

Tutor: Manuel Pineda Sánchez
Cotutor: Rubén Puche Panadero

Agradecimientos

Con la entrega de este trabajo fin de grado, no acabo simplemente un trabajo, sino que pongo fin a una etapa de mi vida.

Esta etapa ha supuesto muchas cosas para mí. No solo me ha formado como ingeniero y profesional, sino que me ha formado como persona, haciendo posible que sepa adaptarme a situaciones que antes de entrar a la universidad parecían imposibles, además de ayudarme a la hora de conocer mis límites para poder superarlos.

Toda esta etapa la he podido superar gracias al esfuerzo y a la perseverancia, pero también debo de admitir que han ayudado mucho mis seres más queridos y cercanos, que aunque no podían ayudarme en cuestiones técnicas o de aprendizaje, estaban ahí para apoyarme y hacer que no decayera. Entre ellos están mis padres, mi hermano, mis amigos de la infancia y amigos de la universidad.

Tengo que agradecer por último a mi tutor Manolo Pineda y a mi cotutor Ruben Puche por guiarme a través de este trabajo y despejarme cualquier duda que me haya surgido.

A todas las personas que he mencionado, os doy mis más sinceras gracias.

Resumen

La finalidad de este trabajo fin de grado es la de realizar un estudio con tal de determinar el mejor sistema para obtener datos de diagnóstico de la centralita de un vehículo, en concreto un Fiat Stilo 1.9 JTD de 2002, para que posteriormente se muestren los datos obtenidos mediante una interfaz gráfica.

Para realizar esta tarea, primero trataremos de conocer el método de comunicación que emplea el vehículo para comunicarse con todos sus controladores.

Habiendo analizado la red del vehículo y sabiendo que este se comunica mediante una red tipo CAN denominada red Mini F.L.Ore.N.C.E, escogeremos una serie de alternativas para compararlas y determinar cuál de ellas es la que mejor se adapta a este proyecto.

Durante este estudio nos decantamos por la primera alternativa seleccionada, el automático, debido a la alta seguridad tanto de su software como de su hardware. A pesar de haber elegido esta alternativa, estudiaremos también el Arduino y la Raspberry como posibles alternativas, que aunque carezcan de tanta seguridad como el automático, son igual de viables y dignas de estudio.

Habiendo analizado las tres alternativas, realizamos una tabla donde se tendrán en cuenta distintos factores relevantes de cada alternativa. Con esto pretendemos hacer más fácil la comparación y también ver los puntos fuertes y débiles de cada alternativa.

Después de esta comparativa, demostramos fehacientemente que el automático es la mejor alternativa, destacando por encima de todo la seguridad que este nos aporta, ya que minimiza cualquier posible fallo en el sistema o en la transmisión de datos.

Finalmente desarrollaremos una simulación con un automático del Departamento de Ingeniería Eléctrica de nuestra universidad, con tal de demostrar las más que sobradas capacidades de esta alternativa, así como el sin fin de posibilidades que el automático nos permite realizar.

Resum

La finalitat d'aquest treball final de grau és la de realitzar un estudi per tal de determinar el millor sistema per obtenir dades de diagnosi de la centraleta d'un vehicle, en concret un Fiat Stilo 1.9 JTD de 2002, perquè posteriorment es mostrin les dades obtinguts mitjançant una interfície gràfica.

Per realitzar aquesta tasca, primer tractarem de conèixer el mètode de comunicació que fa servir el vehicle per comunicar-se amb tots els seus controladors.

Havent analitzat la xarxa del vehicle i sabent que aquest es comunica mitjançant una xarxa tipus CAN anomenada xarxa Mini F.L.Ore.N.C.E, escollirem una sèrie d'alternatives per comparar-les i determinar quina d'elles és la que millor s'adapta a aquest projecte.

Durant aquest estudi ens decantem per la primera alternativa seleccionada, l'autòmata, a causa de l'alta seguretat tant del seu software com del hardware. Tot i havent triat aquesta alternativa, estudiarem també el Arduino i la Raspberry com a possibles alternatives, que encara que no tinguin tanta seguretat com l'autòmata, són igual de viables i dignes d'estudi.

Havent analitzat les tres alternatives, realitzem una taula on es tindran en compte diferents factors rellevants de cada alternativa. Amb això pretenem fer més fàcil la comparació i també veure els punts forts i febles de cada alternativa.

Després d'aquesta comparativa, demostrem fefaentment que l'autòmata és la millor alternativa, destacant sobre tot la seguretat que aquest ens aporta, ja que minimitza qualsevol possible error en el sistema o en la transmissió de dades.

Finalment desenvoluparem una simulació amb un autòmata del Departament d'Enginyeria Elèctrica de la nostra universitat, per tal de demostrar les més que sobrades capacitats d'aquesta alternativa, així com la infinitat de possibilitats que l'autòmata ens permet realitzar.

Abstract

The purpose of this final degree project is to carry out a study in order to determine the best system to obtain diagnosis data from a vehicle's control unit, specifically a Fiat Stilo 1.9 JTD from 2002, for showing the data obtained later through a graphical interface.

To perform this task, we will first try to know the method of communication used by the vehicle to communicate with all its controllers.

Having analyzed the vehicle network and knowing that it is communicating through a CAN type network called the Mini F.L.Ore.N.C.E network, we will choose a series of alternatives to compare them and determine which one is best suited to this project.

During this study we opted for the first alternative selected, the automaton, due to the high security of its software and its hardware. Despite having chosen this alternative, we will also study the Arduino and the Raspberry as possible alternatives, that although they lack as much security as the automaton, are equally viable and worthy of study.

Having analyzed the three alternatives, we made a table where different relevant factors of each alternative will be taken into account. With this we aim to make the comparison easier and also see the strengths and weaknesses of each alternative.

After this comparison, we demonstrate conclusively that the automaton is the best alternative, highlighting above all the security that this gives us, because it minimizes any possible failure in the system or in the transmission of data.

Finally we will develop a simulation with an automaton of the Electrical Engineering Department of our university, in order to demonstrate the more than enough capacities of this alternative, as well as the endless possibilities that the automaton allows us to perform.

Análisis y elección de diferentes sistemas para la obtención de datos de la centralita de un vehículo para su posterior muestra por pantalla

Documento I: Índice general

Febrero de 2018

Universitat Politècnica de València
Grado en Ingeniería Eléctrica

Autor: Carlos Vicente López

Índice general

Documento II: Memoria	8
Documento III: Pliego de condiciones	92
Documento IV: Presupuesto.....	99
Documento V: Anexos	105

Análisis y elección de diferentes sistemas para la obtención de datos de la centralita de un vehículo para su posterior muestra por pantalla

Documento II: Memoria

Febrero de 2018

Universitat Politècnica de València
Grado en Ingeniería Eléctrica

Autor: Carlos Vicente López

Índice

1.	Introducción	15
2.	Objetivos.....	16
3.	Método de comunicación empleado por el vehículo.....	17
3.1.	El avance de la electrónica y comunicación en la automoción	17
3.2.	Puerto OBDII o EOBD	18
3.3.	Bus CAN.....	19
3.4.	Funcionamiento de la electrónica en el vehículo estudiado	21
3.5.	Posibles sistemas para la comunicación	27
4.	Solución alternativa: Autómata programable.....	28
4.1.	Historia de la automática industrial.....	28
4.2.	Elección del autómata	30
Conectividad	31	
Software empleado	32	
Precio.....	32	
Conclusión	33	
4.3.	Conexión e instalación del autómata	34
4.4.	Programación y gestión de datos	37
Configuración de convertidor	37	
Configuración del autómata	40	
4.5.	Visualización de datos a través de pantalla.....	42
Ubicación de pantalla	42	
Selección de pantalla.....	43	
Instalación de pantalla.....	44	
Configuración inicial de la pantalla	45	
Visualización de variables por pantalla	48	
4.6.	Costes y conclusiones	49
5.	Solución alternativa: Arduino.....	51
5.1.	Introducción al Arduino	51
5.2.	Obtención de datos de la centralita	53
Hardware necesario.....	53	
Software empleado	54	
Conexión e instalación.....	55	
5.3.	Visualización de datos a través de pantalla.....	56
Ubicación de pantalla	56	

Selección de pantalla	56
Instalación de pantalla.....	56
Software empleado	57
5.4. Costes y conclusión	58
6. Solución alternativa: Raspberry	60
6.1. Introducción a la Raspberry	60
6.2. Obtención de datos de la centralita	62
Hardware necesario.....	62
Software necesario	65
Instalación de los componentes	66
6.3. Visualización de datos a través de pantalla	66
Ubicación de pantalla	66
Selección de pantalla	67
Instalación de pantalla.....	67
Software empleado	67
6.4. Costes y conclusión	68
7. Solución adoptada	70
8. Implementación de la aplicación.....	72
8.1. Introducción.....	72
8.2. Configuración inicial.....	73
8.3. Funciones de nuestro ejemplo	78
Intermitentes.....	79
Luces	80
Climatización.....	81
8.4. Pantallas de nuestro ejemplo	82
Pantalla Principal	83
Pantalla de la climatización	86
9. Conclusión	90
10. Bibliografía.....	91

Índice de ilustraciones

Ilustración 1: Puerto OBD de un Toyota 4Runner del 1994	17
Ilustración 2: Puerto OBD de la marca BMW hasta 1995	17
Ilustración 3: Puerto OBDII	18
Ilustración 4: Cables trenzados y posibles interferencias que les afectan.....	20
Ilustración 5: Serie bus CAN con resistencias en sus extremos	20
Ilustración 6: Funcionamiento de la red CAN en el Fiat Stilo	21
Ilustración 7: Ubicación de la centralita del habitáculo	22
Ilustración 8: Centralita del habitáculo	22
Ilustración 9: Estructura de la red Mini F.L.Ore.N.C.E.....	23
Ilustración 10: Estructura de la red C-CAN	24
Ilustración 11: Estructura de la red B-CAN	25
Ilustración 12: Estructura de la línea A Bus	25
Ilustración 13: Estructura de la línea K.....	26
Ilustración 14: Pines del conector EOBD	26
Ilustración 15: Primer autómat, creado por Arquitas de Tarento.....	28
Ilustración 16: Regulador de Watt.....	28
Ilustración 17: Emblema de la ETSID	29
Ilustración 18: Siemens 6ES7212-1HE40-0XB0	30
Ilustración 19: Schneider Electric TM221CE16R.....	31
Ilustración 20: Omron CP1L-L10DT-D.....	31
Ilustración 21: Precio del autómat de Omron	33
Ilustración 22: Precio del autómat de Schneider	33
Ilustración 23: Precio del autómat de Siemens	33
Ilustración 24: Pines del conector EOBD	34
Ilustración 25: Conector macho puerto EOBD	35
Ilustración 26: Correo recibido de www.adfweb.com	35
Ilustración 27: Transformador de 12 V a 24 V.....	36
Ilustración 28: Menú principal del SW67606	37
Ilustración 29: Menú "Set Communication" del SW67606	38
Ilustración 30: Menú "Recieve Frames" del SW67606.....	38
Ilustración 31: Menú "Info Recieve" del SW67606	39
Ilustración 32: Menú "Send Frames" del SW67606	39
Ilustración 33: Menú "Info Send" del SW67606.....	40
Ilustración 34: Selección de autómat en Tia Portal V14.....	40
Ilustración 35: Búsqueda de archivo xml mediante Tia Portal V14.....	41
Ilustración 36: Instalación de archivo xml en Tia Portal V14	41
Ilustración 37: Búsqueda de convertidor en el Tia Portal V14.....	41
Ilustración 38: Conexión entre autómat y convertidor en Tia Portal V14	42
Ilustración 39: Asignar nombre al convertidor mediante Tia Portal V14.....	42
Ilustración 40: Asignar dirección ip mediante Tia Portal V14	42
Ilustración 41: Zona central del salpicadero del Fiat Stilo.....	43
Ilustración 42: Modelo de HMI escogido	43
Ilustración 43: Modelo de tablet escogido.....	44
Ilustración 44: Transformador de 12 V a 5 V.....	44
Ilustración 45: Switch de puertos Ethernet.....	45

Ilustración 46: Adaptador Ethernet a micro-usb.....	45
Ilustración 47: Ladrón de puertos micro-usb	45
Ilustración 48: Insertar tablet en el Tia Portal V14.....	46
Ilustración 49: Creación de puerto de red para la tablet mediante Tia Portal V14	46
Ilustración 50: Esquema y direcciones ip que componen la red	47
Ilustración 51: Acción "Cargar en dispositivo " en Tia Portal V14.....	47
Ilustración 52: Menú principal del programa Win CC Runtime Loader	47
Ilustración 53: Agregar imágenes y plantillas mediante Tia Portal V14.....	48
Ilustración 54: Escoger resolución de la pantalla del Tia Portal V14.....	48
Ilustración 55: Opciones de personalización de un indicador gauge	49
Ilustración 56: Prototipo del primer Arduino	51
Ilustración 57: Modelo actual de placa Arduino	52
Ilustración 58: Placa Arduino UNO	52
Ilustración 59: Precio de placa modelo Arduino UNO Rev3.....	53
Ilustración 60: Shield BUS CAN modelo MakerHawk	54
Ilustración 61: Cable adaptador ODB2 a DB9.....	55
Ilustración 62: Conector Dc 2.1	55
Ilustración 63: Pantalla modelo 971-GEN4-ULCD-70DT.....	56
Ilustración 64: Shield adaptador para pantalla	57
Ilustración 65: Tarjeta microSD	57
Ilustración 66: Prototipo de la primera Raspberry	60
Ilustración 67: Esquema de la Raspberry Pi 3 Model B	61
Ilustración 68: Raspberry Pi 3 Model B	62
Ilustración 69: Raspberry junto con la Shield PiCAN 2	63
Ilustración 70: Pines del conector EOBD	63
Ilustración 71: Transformador de 12 V a 5 V	64
Ilustración 72: Conector macho puerto EOBD	64
Ilustración 73: Cable micro USB	64
Ilustración 74: Tarjeta MicroSD	65
Ilustración 75: Pantalla Makibes 7 pulgadas	67
Ilustración 76: Autómata del laboratorio	72
Ilustración 77: Opción "Agregar dispositivo" en Tia Portal	73
Ilustración 78: Selección del autómata en Tia Portal	74
Ilustración 79: Selección de la pantalla en Tia Portal.....	75
Ilustración 80: Añadido de una conexión red a la pantalla mediante Tia Portal	75
Ilustración 81: Conexión física entre el autómata y el portátil.....	76
Ilustración 82: Conexión entre autómata y portátil mediante Tia Portal	76
Ilustración 83: Configuración de dirección Ip mediante Tia Portal	76
Ilustración 84: Cargado del programa al autómata mediante Tia Portal.....	77
Ilustración 85: Búsqueda del autómata mediante Tia Portal.....	77
Ilustración 86: Carga de programa en el autómata.....	78
Ilustración 87: Flujograma de la función Intermitentes.....	79
Ilustración 88: Bloque de la función Luces	80
Ilustración 89: Primer Flujograma de la función Climatización.....	81
Ilustración 90: Segundo flujograma de la función Climatización	82
Ilustración 91: Pantalla principal de nuestro ejemplo	83
Ilustración 92: Acceder a tabla de variables de la pantalla.....	84

Ilustración 93: Ajustar la escala de una variable	84
Ilustración 94: Visualización de las luces mediante la pantalla principal.....	85
Ilustración 95: Indicador de pantalla y botón de cambio de pantalla.....	85
Ilustración 96: Pantalla secundaria de nuestro ejemplo	86
Ilustración 97: Cuentarrevoluciones horizontal	86
Ilustración 98: Cuentakilómetros numérico	87
Ilustración 99: Visualización de las luces mediante la pantalla secundaria	87
Ilustración 100: Primera parte del control de la climatización	88
Ilustración 101: Segunda parte del control de la climatización	88
Ilustración 102: Indicador de pantalla y botón de cambio de pantalla.....	89

Índice de tablas

Tabla 1: La velocidad de transferencia de datos frente a la longitud del bus CAN.....	20
Tabla 2: Características de los autómatas programables.....	31
Tabla 3: Comparativa de los autómatas programables.....	34
Tabla 4: Costes de la primera alternativa, autómata industrial.....	49
Tabla 5: Valoración de la alternativa autómata industrial.....	50
Tabla 6: Costes de la segunda alternativa: Arduino.....	58
Tabla 7: Valoración de la alternativa Arduino.....	59
Tabla 8: Coste de la tercera Alternativa: Raspberry.....	68
Tabla 9: Valoración de la alterativa Raspberry.....	69
Tabla 10: Comparativa de las tres alternativas estudiadas.....	70
Tabla 11: Costes de la alternativa escogida, autómata programable.....	101
Tabla 12: Coste de las horas empleadas en el proyecto.....	101
Tabla 13: Coste de licencias o software contratados.....	102
Tabla 14: Coste de herramientas empleadas.....	102
Tabla 15: Coste neto.....	103
Tabla 16: Margen de beneficio.....	103
Tabla 17: Coste total del proyecto.....	104

1. Introducción

La humanidad se ha movido desde sus primeros pasos por el deseo de saber y dar una explicación a todo aquello que desconoce. Es por esto que hemos avanzado tanto y tan rápido a lo largo de nuestra existencia.

Este avance ha sido posible gracias a inventos como la rueda, que nos ayudó a transportar mercancías y hacer más fácil la construcción. También está la imprenta, que nos facilitaba la comunicación y el dejar nuestros conocimientos guardados para el estudio de posteriores generaciones. Hay que nombrar también al automóvil, el cual se comenzó a crear a finales del siglo XVIII y que supuso un cambio en la sociedad a partir del siglo XIX. Por último hay que mencionar el que es sin duda el último y gran descubrimiento de la humanidad, la computación, que ha supuesto una revolución en nuestro más reciente pasado, y sin la cual ahora mismo sería muy difícil imaginar nuestro día a día.

En este Trabajo Fin de Grado (ahora en adelante TFG) vamos a hablar y explicar cómo se unen los dos últimos inventos antes nombrados, el automóvil y la computación.

Gracias a esta unión podemos mejorar numerosas características de los automóviles, desde su velocidad, hasta su seguridad, pasando por su eficiencia. Esto es posible dado que se han sustituido muchísimos componentes mecánicos por electrónicos, lo que nos aporta mayor seguridad, control y fiabilidad.

Un ejemplo de esta mejoría es el sensor de presión de las ruedas, que nos indica en todo momento por un display la presión de los neumáticos del vehículo. Esto nos aporta seguridad a la hora de detectar un pinchazo a tiempo y nos ahorra el tener que estar comprobando la presión con bombas manuales.

Dicho esto, el vehículo sobre el que vamos a trabajar es un Fiat Stilo 1.9 JTD de 2002. Aunque como explicaremos más adelante, este estudio podría ser utilizado en la mayoría de vehículos de la actualidad.

2. Objetivos

El objetivo principal de este TFG es el de realizar un análisis de los métodos de comunicación que emplea el vehículo escogido para comunicarse e intercambiar datos con los distintos elementos y sensores del mismo vehículo y con otros elementos externos, como pueden ser aparatos de diagnóstico, PLCs o controladores.

Una vez realizado este análisis y conociendo el protocolo o método de comunicación del vehículo, se procederá a nombrar tres opciones que permitan comunicarnos con el vehículo y mostrar los datos que deseemos mediante una pantalla que posteriormente se ubicaría en el panel frontal del vehículo.

Primero de todo analizaremos la viabilidad de realizar la tarea que deseamos mediante un autómata programable, y posteriormente la viabilidad de la misma mediante un Arduino y una Raspberry. Al mismo tiempo que estudiamos cada alternativa realizaremos las configuraciones iniciales para poner en funcionamiento cada una de estas.

Habiendo visto las tres alternativas que tenemos, procederemos a compararlas gracias a toda la información que tenemos de cada una, destacando por encima de todo la seguridad que nos aporta cada una de las alternativas.

Finalmente mostraremos ejemplos reales de la cantidad de posibilidades que nos ofrece la alternativa del autómata programable. Para ello emplearemos las herramientas que nos facilita la UPV (Universitat Politècnica de València) en su Departamento de Ingeniería Eléctrica. Con esto pretendemos convencer de que lo que se explica en este TFG es factible y ofrece muchas posibilidades.

Mediante esta investigación no solo determinaremos cual es la mejor forma de sustraer información del Fiat Stilo para mostrarla mediante una pantalla, sino que nos servirá para la gran mayoría de vehículos del mercado. Esto se debe principalmente a que el sistema de comunicación que emplean los vehículos actuales está estandarizado, como veremos a lo largo de este TFG.

3. Método de comunicación empleado por el vehículo

3.1. El avance de la electrónica y comunicación en la automoción

El primer motivo por el que la electrónica se introdujo en el mundo del automóvil, fue por raro que parezca, por motivos medioambientales.

Y es que durante los años 60 se produjo un “boom” en el sector del automóvil y las ventas de estos crecieron mucho, lo que conllevó a que la contaminación comenzara a ser un problema en los grandes centros de población, dado que no existía ninguna medida de control de emisiones y que los sistemas de combustión de los vehículos eran muy poco eficientes.

A finales de los años 80 se comenzó a introducir la electrónica en el sistema de inyección de los vehículos, de forma que mediante sensores y una centralita básica se logró utilizar la cantidad justa de combustible para cada ciclo del motor, consiguiendo así un ahorro en combustible y una menor emisión de gases tóxicos a la atmosfera.

Por lo tanto la primera toma de contacto entre la electrónica y el automóvil fue una simple centralita de control de inyección. Y esto a pesar de ser un gran avance supuso un pequeño inconveniente, la compatibilidad, ya que cada fabricante creaba sus propias centralitas, protocolos y modos de comunicación. Esto provocó que el organismo CARB (California Air Resources Board) regulara y unificara todos estos métodos de comunicación en los Estados Unidos de América a partir del año 1988, creando así un método estandarizado conocido como OBD (On Board Diagnostic), del cual hablaremos más adelante, en el apartado “3.4. Puerto OBDII o EOBD”.



Ilustración 1: Puerto OBD de un Toyota 4Runner del 1994



Ilustración 2: Puerto OBD de la marca BMW hasta 1995

Pero esta unión entre la electrónica y el automóvil siguió evolucionando sin parar, aumentándose la cantidad de elementos electrónicos por todo el vehículo, llegando a haber múltiples centralitas por todo el vehículo, haciéndose cargo cada una de funciones concretas. Sin ir más lejos un ejemplo son la centralita que se encarga de todas las funciones de la puerta del conductor, otra que se encarga de la climatización y otra que controla la inyección del motor.

A pesar de haber múltiples centralitas por vehículo, todas estas están conectadas por una red de comunicaciones que funciona bajo el protocolo CAN (Controller Area Network), que explicaremos en el siguiente apartado. Gracias a esta red las numerosas centralitas pueden compartir información entre sí y comunicarse con elementos de diagnóstico externos.

Actualmente es tanta la unión entre ambos sectores, que la marca Tesla Inc ha sido la primera marca en comercializar un vehículo de propulsión eléctrica y con piloto automático. Este piloto automático está controlado gracias a un gran número de sensores, cámaras y radares a bordo. Estos envían información a la centralita principal del coche, que interpreta toda la información recibida, para que el vehículo haga las maniobras oportunas en cada momento. Y es tanta la conectividad hoy en día, que podemos obtener datos de nuestro vehículo desde nuestro teléfono móvil o desde casa.

3.2. Puerto OBDII o EOBD

A continuación veremos cómo y mediante qué sistemas se comunica el vehículo con elementos externos, ya sean sistemas de diagnóstico, ordenadores, o cualquier otro sistema.

Partimos de la base de que toda la toma de datos de los vehículos americanos se realizaba mediante el sistema OBD a finales de los años 80, y de que estos datos se tomaban básicamente para saber si los vehículos no emitían gases contaminantes por encima de los límites estipulados por la CARB.

Poco después, a partir de 1995 se comenzó a emplear el protocolo OBDII, exigido tanto por la CARB como por la EPA (Environmental Protection Agency). De esta forma se mejorarían las capacidades del sistema OBD y se incrementaría el control sobre la emisión de gases de efecto invernadero.

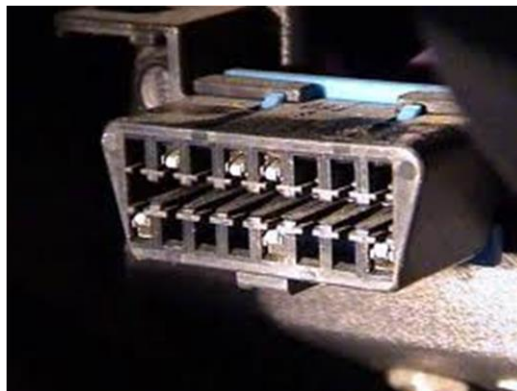


Ilustración 3: Puerto OBDII

Por tanto el sistema OBDII se emplea a partir de esa fecha en todo tipo de vehículos a motor, aunque sean híbridos. Este sistema controla los sistemas o sensores de emisiones y todos los componentes que puedan afectar directa o indirectamente a los gases de escape u otro tipo de contaminantes. Debido a esto, si algún componente no trabajaba correctamente o registra una cantidad superior a la normal de emisión de gases de efecto invernadero, se emite y almacena un DTC (Diagnostic Trouble Problem) y se emite un aviso para que el conductor sea consciente del fallo que se ha producido.

El aviso que recibe el conductor se emite cuando han transcurrido dos ciclos de conducción desde que se ha detectado un DTC, evitándose así falsos errores del sistema. Este aviso cesará una vez hayan transcurrido 3 ciclos de conducción desde que se detectó por última vez el DTC.

Un ciclo de conducción se trata de un análisis de todos los sistemas del OBDII del vehículo mientras se simulan distintos tipos de conducción cotidianas. Estos ciclos se realizan normalmente a unos 25 °C, en naves cerradas (sin resistencia al aire) y sobre rodillos que simulan la inercia propia del vehículo.

Una vez se ha detectado un DTC, este será borrado de la memoria cuando hayan transcurrido 40 ciclos de arranque y calentamiento del motor desde que se apagó en aviso al conductor.

En el momento de producirse un DTC, se almacenan todos los datos telemétricos del vehículo para que durante la inspección previa a la reparación, se pueda determinar con exactitud qué fue lo que provocó el fallo. En caso de varios DTCs, se almacenan los datos únicamente del último.

Debido al gran avance que supuso todo el sistema OBDII, Europa desarrolló en el año 2000 el EOBD (European On Board Diagnostic), que a grandes rasgos es muy parecido al OBDII. De esta forma Europa obligó a los fabricantes europeos a instalar en sus vehículos sistemas de diagnóstico como los que ya poseían los vehículos americanos. Este sistema se comienza a aplicar en el año 2000 para vehículos propulsados mediante gasolina y en 2003 para vehículos propulsados mediante gasoil.

Se crea entonces una estandarización donde tanto el OBDII como el EOBD emplean sistemas de diagnóstico, interfaces y conectores muy similares. Por tanto el análisis de los sistemas y la detección de fallos están normalizados y son muy fáciles y económicos de realizar, sea cual sea la marca y modelo del vehículo.

Debido a esto, se estipula que según la norma ISO 15031-3:2004 el conector OBDII o EOBD deben de estar ubicados en el habitáculo del vehículo, concretamente detrás del cenicero, bajo el panel de instrumentos o junto con el controlador del habitáculo. En el caso del vehículo que se está estudiando, el conector se encuentra junto con el controlador del habitáculo y la caja de fusibles, bajo el panel de instrumentos.³

3.3. Bus CAN

El bus CAN (Controller Area Network) se trata de un protocolo de comunicaciones desarrollado por la empresa Robert Bosch GmbH en 1986. En 1988 se empleó este protocolo por primera vez en un vehículo, en el BWM serie 8. En 1991 se desarrolló el CAN 2.0 que emplea identificadores de 29 bits, en vez de los 11 bits de la versión anterior. El bus CAN se estandarizó gracias a la norma ISO (International Organization for Standardization) 11898 en 1993. Finalmente en 2011 Bosch comenzó a desarrollar junto con otros expertos el CAN-FD (flexible data-rate), que es capaz de transmitir datos por encima del límite de 1 Mbps de su predecesor.

Cabe destacar que aunque el automovilismo se ha beneficiado en gran medida de este tipo de bus, también ha sido muy importante en el mundo de la industria. Se ha logrado

conectar toda la maquinaria y procesos de las fábricas entre sí para que trabajen de una forma mucho más eficientes. Gracias a esto por ejemplo podemos obtener datos de la producción de una máquina sin necesidad de estar presente en la misma sala que esta y automáticamente regular el resto de la producción para que esta máquina esté en continuo funcionamiento.

Físicamente la transmisión entre buses CAN se realiza generalmente en serie mediante dos cables trenzados entre sí. El trenzado se efectúa para reducir las emisiones electromagnéticas y las interferencias inducidas en el bucle que forman ambos cables.

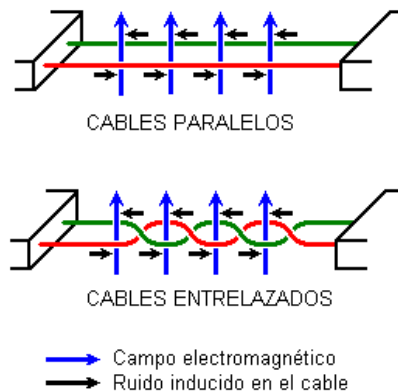


Ilustración 4: Cables trenzados y posibles interferencias que les afectan

También se colocan resistencias al principio y al final de la línea de comunicaciones. Estas resistencias tienen que tener un valor igual a la impedancia propia de la línea. Esto neutralizará las posibles reflexiones de energía en los extremos de la línea, que distorsionan o incluso eliminan la transmisión de datos a través de esa línea.

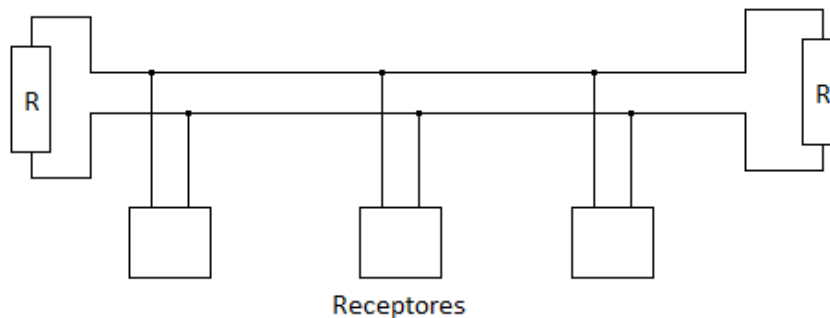


Ilustración 5: Serie bus CAN con resistencias en sus extremos

En cuanto a longitud de la serie se refiere, existe un límite orientativo que relaciona la tasa de kbits por segundo con la longitud del cable, llegando a ser la longitud máxima del cable de 1200 m. Esta regla viene dada por la siguiente tabla.

Tasa de transferencia (kbits/s)	Longitud de la serie (m)
1000	40
500	100
250	200
100	500
50	1000

Tabla 1: La velocidad de transferencia de datos frente a la longitud del bus CAN

La comunicación en el bus CAN se efectúa mediante dos canales, que corresponden a los dos cables trenzados, denominándose estos CAN_H (CAN High) y CAN_L (CAN Low). La señal que transmiten estos canales es de tipo Voltaje, donde cada canal envía un determinado voltaje, para que luego el receptor de la señal calcule la diferencia entre ambos voltajes. Esta diferencia será traducida a código binario para poder comprender la información que se está transfiriendo. A continuación exponemos un ejemplo de cómo funciona este sistema en el vehículo que vamos a estudiar, el Fiat Stilo.

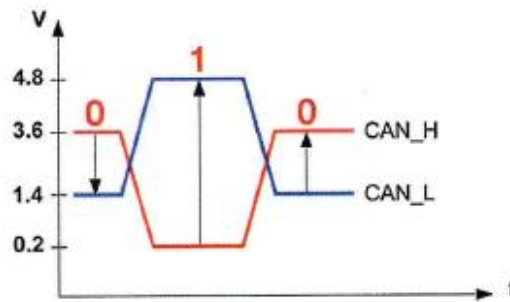


Ilustración 6: Funcionamiento de la red CAN en el Fiat Stilo

$$\text{CAN}_H - \text{CAN}_L = 3.6 - 1.4 = 2.2 \text{ V (bit 0)}$$

$$\text{CAN}_H - \text{CAN}_L = 4.8 - 0.2 = 4.6 \text{ V (bit 1)}$$

Ahora que ya sabemos cómo se transmiten los bits a través de la red, vamos a ver cómo se realiza la comunicación gracias a estos bits.

La comunicación se efectúa mediante mensajes, los cuales están formados por bloques bits, que ordenan de la siguiente forma:

- Bit de inicio: Este bit va el primero de todos e indica que se está transmitiendo un mensaje. Su estado es siempre "0".
- Identificador: Este bloque tiene 11 o 29 bits de extensión, dependiendo de si el formato del mensaje es extendido o no. Este nos indica el identificador al igual que la prioridad del receptor o nodo.
- Código de longitud de datos: Aquí se nos informará mediante una serie de 6 bits la longitud del mensaje que vamos a transmitir.
- Campo de datos: Este bloque puede tener una longitud de hasta 8 bytes (64bits). Aquí vienen contenidos los datos del mensaje que se está enviando.
- Campo de verificación por redundancia cíclica (CRC): Este bloque está formado por 15 bits y su función es la de detectar errores en la transmisión del mensaje.
- Fin de trama: se trata de una sucesión de 7 bits que indican el final del mensaje.

3.4. Funcionamiento de la electrónica en el vehículo estudiado

Una vez nos hemos introducido en el mundo de la electrónica en el automóvil y explicado cómo se comunica esta electrónica a través de la red CAN, vamos a explicar detalladamente como está formada la arquitectura electrónica del vehículo que tomaremos por ejemplo, el Fiat Stilo 1.9 JTD de 2002.

En este vehículo se ha implementado una red de multiplexado denominada Mini F.L.Ore.N.C.E (Fiat Luxury-car OrientEd Networks and Control Electronics). Esta red se

encarga de comunicar todos los calculadores del vehículo con el calculador o centralita del habitáculo, ubicado debajo del panel de instrumentos del vehículo.



Ilustración 7: Ubicación de la centralita del habitáculo

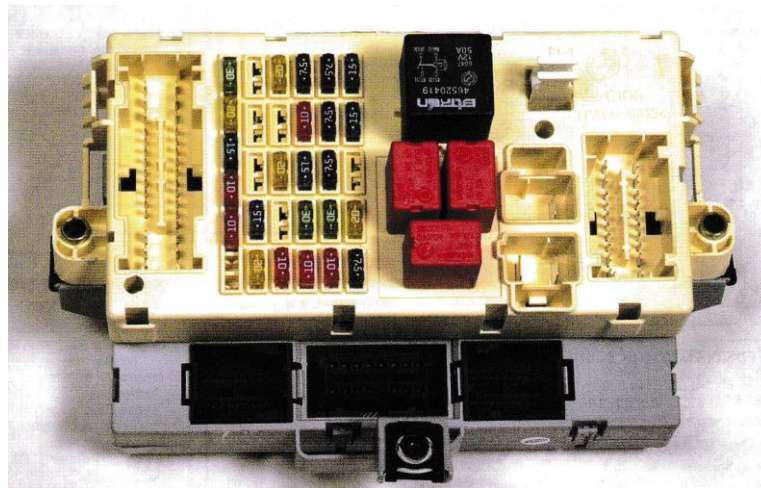


Ilustración 8: Centralita del habitáculo

La red Mini F.L.Ore.N.C.E está compuesta a su vez por un total de 2 subredes y 3 líneas. Existen dos subredes principales como son la B-CAN (baja velocidad) y la C-CAN (alta velocidad), y después las líneas K, W y A Bus. A continuación se muestra un esquema de la red del vehículo y de todas las redes y controladores que lo componen.

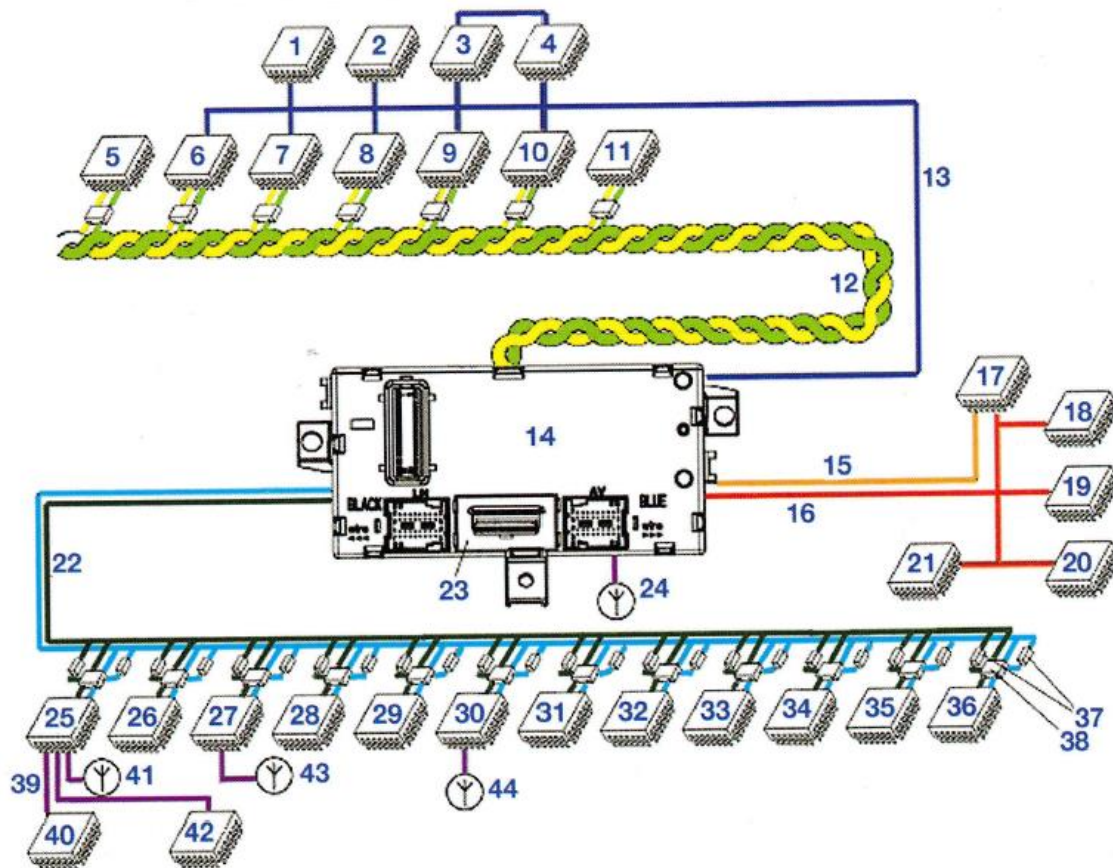


Ilustración 9: Estructura de la red Mini F.L.Ore.N.C.E

- | | | |
|-------------------------------------|--|--|
| 1-Calculador de suspensión | 18-Calculador de alarma antirrobo | 32-Calculador de asiento delantero derecho |
| 2-Calculador de airbags | 19-Grupo de interruptores | 33-Calculador de puerta conductor |
| 3-Calculador luz derecha | 20-Captador de lluvia | 34-Calculador de puerta pasajero |
| 4-Calculador luz izquierda | 21-Alarma volumétrica | 35-Calculador volante |
| 5-Selector de velocidad | 22- Red B-CAN (baja velocidad) | 36-Cuadro de instrumentos |
| 6-Control velocidad de cruceo | 23-Toma de diagnóstico | 37-Conexiones |
| 7-Calculador dirección asistida | 24-Antena antiarranque | 38-Terminador CAN |
| 8-Calculador de caja de velocidad | 25-Autoradio | 39-Audio multiplexado |
| 9-Calculador de gestión motor | 26-Calculador de climatización | 40-Cargador CD |
| 10-Calculador ABS | 27-Calculador de apertura | 41-Antena GPS |
| 11-Captador de ángulo volante | 28-Calculador compartimento equipajes | 42-Amplificador autoradio |
| 12-Red C-CAN (alta velocidad) | 29-Calculador de bloqueo de columna de dirección | 43-Antena Easy Go |
| 13-Línea K (diagnóstico) | 30-Calculador de ayuda al estacionamiento | 44-Radar de estacionamiento |
| 14-Calculador habitáculo | 31-Calculador de asiento delantero izquierdo | |
| 15-Línea K (diagnóstico) | | |
| 16-Línea A Bus | | |
| 17-Calculador de presión neumáticos | | |

Cómo podemos observar a simple vista, las distintas redes comunican todos los controladores con el controlador del habitáculo o centralita. De esta manera se transmite toda la información a la centralita, para que esta la procese y se pueda posteriormente reenviar esta información procesada si se requiere a los distintos controladores.

Dentro de esta red cada red y cable tiene una función distinta:

- La **red C-CAN** (alta velocidad) transmite la información en código binario como hemos visto anteriormente, de forma que está compuesta por el cable CAN-H de color rosa y negro y por el cable CAN-L de color rosa y blanco.

Esta red, debido a los voltajes que emplea para transmitir la información en código binario, pierde toda capacidad de transmitir o recibir información si cualquiera de sus dos cables es cortado, entra en cortocircuito con el otro cable, con el polo positivo de la batería, o con la masa.

Al tratarse de la red de alta velocidad, se encarga de comunicar los calculadores más importantes, como son los que controlan el motor o las medidas de seguridad.

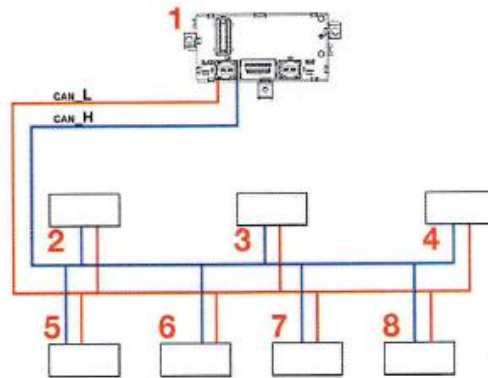


Ilustración 10: Estructura de la red C-CAN

1-Calculador habitáculo
2-Control velocidad de cruceo adaptable

3-Calculador dirección asistida
4-Calculador caja automática
5- Calculador de gestión motor

6- Calculador ABS
7-Selector de velocidad
8-Captador de ángulo volante

- La **red B-CAN** (baja velocidad) transmite información al igual que la red C-CAN, mediante código binario y dos cables, el cable CAN-B de color rosa y negro y por el cable CAN-A de color rosa y blanco.

Debido a que los voltajes que emplea esta red para comunicarse en código binario son distintos a los de la red C-CAN, a pesar de un fallo en los cables esta puede seguir transmitiendo información y detectar anomalías globales y locales por todo el sistema. Los fallos que esta red soportan son el corte de uno de sus cables, el cortocircuito de uno de estos con el otro cable, con el polo positivo de la batería, o con la masa. Por lo tanto esta red en el peor de los casos se puede comunicar de forma unifilar, aunque si se cortan ambos cables o se produce un cortocircuito de ambos cables con la masa se imposibilita la comunicación.

Al tratarse de la red de baja velocidad, se encarga de la comunicación de los calculadores secundarios, que básicamente nos aportan confort a la hora de conducir pero ninguna protección o medida de seguridad. Un ejemplo de estos sería el elevavinas eléctrico, la radio o la climatización.

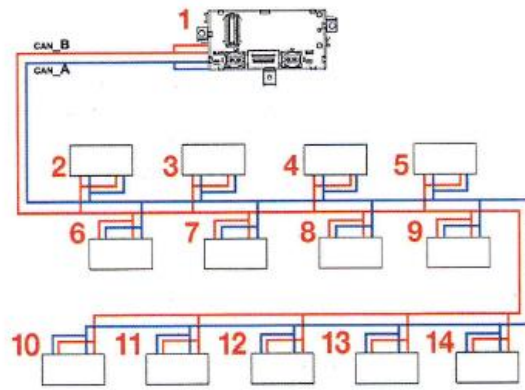


Ilustración 11: Estructura de la red B-CAN

- | | | |
|---|-----------------------------------|------------------------------------|
| 1-Calculador habitáculo | 6-Volante (mando radio y connect) | 12-Calculador de asiento conductor |
| 2-Calculador Easy Go | 7-Cuadro de instrumentos | 13-Calculador de asiento pasajero |
| 3-Calculador de puerta conductor | 8-Calculador de climatización | 14-Calculador de estacionamiento |
| 4-Calculador de puerta pasajero | 9-Calculador de climatización | |
| 5-Calculador de bloqueo de columna de dirección | 10-Connect | |
| | 11-Autoradio | |

- La **línea A Bus** es diferente frente a las redes CAN que ya hemos explicado. Esta línea se hace cargo de los sensores de alarma y de los captadores volumétricos.

La transmisión de datos se realiza a una velocidad de 4,8 Kbit/s y de forma que cada vez que se desea transmitir información, el calculador de origen debe de especificar la dirección del calculador de destino en el bloque de datos. La transmisión además es monodireccional de forma que el calculador receptor de la información debe responder al emisor para concluir la transmisión de datos.

Esta línea se encarga de comunicar el calculador del habitáculo con la sirena antirrobo, las alarmas volumétricas, el grupo de interruptores, el captador de lluvia y luminosidad, y el calculador de presión de los neumáticos.

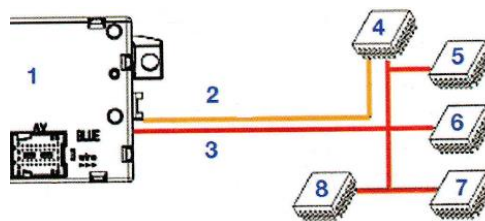


Ilustración 12: Estructura de la línea A Bus

- 1-Calculador habitáculo
- 2-Línea K (diagnóstico)
- 3-Línea A Bus
- 4-Calculador de presión neumáticos
- 5-Calculador de alarma antirrobo
- 6-Grupo de interruptores
- 7-Captador de lluvia
- 8-Alarma volumétrica

- La **línea K** permite el diagnóstico mediante el aparato de diagnosis de todos los calculadores que están comunicados con el calculador del habitáculo mediante la red C-CAN y también de los calculadores de suspensión airbags, de luz derecha y de luz izquierda.

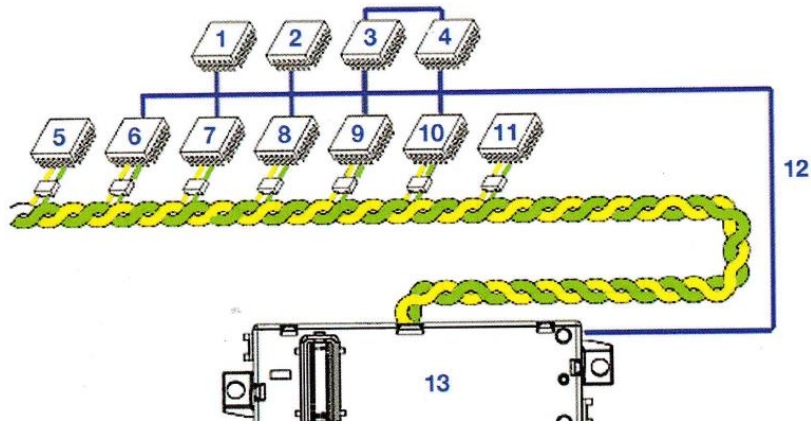
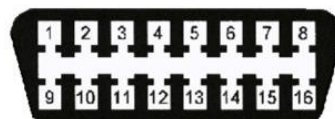


Ilustración 13: Estructura de la línea K

- | | | |
|----------------------------|-----------------------------------|-------------------------------|
| 1-Calculador de suspensión | 6-Control velocidad de cruceo | 11-Captador de ángulo volante |
| 2-Calculador de airbags | 7-Calculador dirección asistida | 12-Línea K (diagnóstico) |
| 3-Calculador luz derecha | 8-Calculador de caja de velocidad | 13-Calculador habitáculo |
| 4-Calculador luz izquierda | 9-Calculador de gestión motor | |
| 5-Selector de velocidad | 10-Calculador ABS | |

- La **línea W** se trata de una línea independiente del resto, cuya función es la de comunicar el calculador del habitáculo con el calculador de gestión del motor. Por tanto se trata de una línea destinada totalmente a la seguridad, ya que si se produce un corte de la red C-CAN, la comunicación entre los controladores nombrados anteriormente sigue en pie.

Analizada toda la red del Fiat Stilo, vamos a centrarnos en el puerto EOBD del vehículo, dado que nos facilita el acceso a la mayoría de redes y líneas que forman la red Mini F.L.Ore.N.C.E del Fiat Stilo. A continuación mostramos la función de cada uno de los pines del conector:



- | | |
|--------------------------|---------------------------|
| 1 - Sin uso | 9 - Sin uso |
| 2 - J1850 Bus positivo | 10 - J1850 Bus negativo |
| 3 - Sin uso | 11 - Sin uso |
| 4 - Tierra del Vehículo | 12 - Sin uso |
| 5 - Tierra de la Señal | 13 - Tierra de la señal |
| 6 - CAN High | 14 - CAN Low |
| 7 - ISO 9141-2 - Línea K | 15 - ISO 9141-2 - Línea L |
| 8 - Sin uso | 16 - Batería - positivo |

Ilustración 14: Pines del conector EOBD

Como vemos en la imagen, mediante este conector podremos tener acceso a las redes CAN y las líneas K y L del vehículo. Por lo tanto esta será la mejor forma de obtener la información que deseemos del vehículo estudiado, además de que tenemos un pin de +12 V y uno de masa directos desde la batería con los que podremos suministrar corriente a los elementos que queramos instalar.

3.5. Posibles sistemas para la comunicación

Llegados a este punto, ya conocemos como se comunican entre sí todos los controladores del vehículo mediante el protocolo Bus CAN y también conocemos un medio para comunicarnos con la red del Fiat Stilo como es el puerto EOBD.

Ahora queda ver que herramientas o sistemas vamos a emplear para comunicarnos con la red del vehículo a través del puerto EOBD, buscando también que sea capaz de manejar la información que substraigamos de la red. Vamos a plantear por tanto 3 posibles soluciones alternativas que sean capaces de realizar esta tarea.

Enumeramos a continuación una serie de requisitos a cumplir por cada una de las alternativas:

- Debe de ser capaz de comunicarse mediante una red CAN
- Debe de ser de tamaño contenido para poder ubicarla dentro del Fiat Stilo
- El consumo eléctrico del sistema no debe ser muy elevado dado que queremos suplirnos de la electricidad suministrada por la batería del vehículo
- El precio debe de ser contenido y no superar los 1000 € de costes en materiales
- Debe de permitirnos mostrar la información substraída de la red del vehículo a través de una pantalla o interfaz gráfica

Habiendo visto estos requisitos, vamos a nombrar un total de 3 sistemas que nos permitirán hacer la tarea deseada cumpliendo con todos los requisitos antes expuestos:

- Autómata programable
- Arduino
- Raspberry

Estos sistemas nos permitirán establecer una comunicación con la red CAN del Fiat Stilo y a la vez serán capaces de manejar la información de esta red para poder mostrarla a través de una interfaz gráfica o pantalla.

Durante los siguientes apartados, estudiaremos detenidamente cada una de las soluciones alternativas, así como determinaremos su coste y explicaremos como realizar las configuraciones iniciales de cada una de las alternativas.

Después de analizar cada solución alternativa, las compararemos y elegiremos una de estas para realizar una implementación de la alternativa a modo de ejemplo. Cabe destacar que en la comparativa analizaremos los aspectos más relevantes de cada una, dándole especial importancia a la seguridad que nos aporte cada solución alternativa.

4. Solución alternativa: Autómata programable

4.1. Historia de la automática industrial

Antes de introducirnos en la historia de la automática, vamos a preguntarnos primero que es en sí la automática. Según la Real Academia de las Ciencias la automática es *la Disciplina que trata de métodos y procedimientos cuya finalidad es la sustitución del operador humano por un operador artificial en la ejecución de una tarea física y mental previamente programada*. Podemos deducir por tanto que la automática tiene como objetivo sustituir la acción o decisión humana para la realización de tareas.¹

Se dice que el primer autómata industrial data del siglo IV a.C. Fue creado por el filósofo y matemático Arquitas de Tarento (430 a.C - 360 a.C), octavo sucesor de Pitágoras. Este inventó una máquina voladora con forma de paloma, que con alas articuladas, equilibrada mediante pesos y gracias a un motor interior de vapor consiguió desplazarse volando cerca de 300 metros.

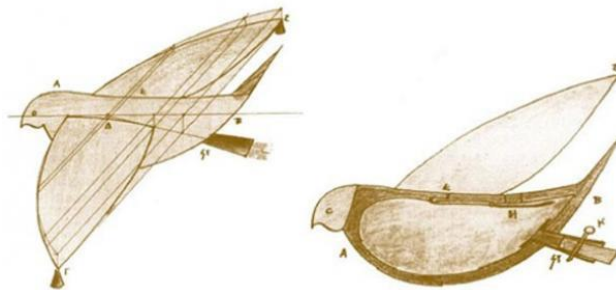


Ilustración 15: Primer autómata, creado por Arquitas de Tarento

Otro de los grandes inventos de la automática es el regulador centrífugo, también conocido como el regulador de Watt. Este regulador fue desarrollado por James Watt (1736 - 1819) en 1788, aunque se conoce que los molinos de viento a partir del siglo XVII ya llevaban equipado este sistema.

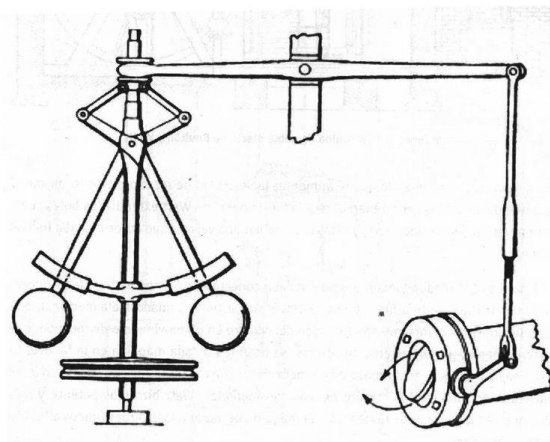


Ilustración 16: Regulador de Watt

Cómo podemos apreciar en la ilustración, el sistema consta de una polea, dos contrapesos, una barra móvil y una válvula de alimentación de motor o un freno que actúa sobre la aspas de un molino.

El sistema funciona según la velocidad que se le transmita desde la polea, estando conectada esta al motor o a las aspas del molino. Esta velocidad causa un aumento de la fuerza centrípeta, lo que conlleva a un alejamiento de los contrapesos respecto del eje. Debido a este alejamiento la barra móvil se desplaza, lo que conlleva finalmente a la apertura/cierre de la válvula que alimenta el motor o al frenado de las aspas del molino.

Como curiosidad cabe destacar que debido a la importancia de esta invención, esta forma parte del emblema de nuestra escuela, la Escuela Técnica Superior de Ingeniería del Diseño.



Ilustración 17: Emblema de la ETSID

Después de esto la automática siguió avanzando poco a poco hasta que a principios del siglo XX se comenzó a dar utilidad a la electricidad de formas que antes ni se imaginaban, creándose por ejemplo la televisión y la radio. Gracias a los avances realizados hasta la fecha Henry Ford propulsó enormemente el mundo del automóvil empleando la novedosa producción en cadena para crear su automóvil, el Ford T.

Pero el avance de la automática industrial no había hecho más que comenzar. A mediados del siglo XX surgió la electrónica, y con esta la posibilidad de administrar información mediante circuitos electrónicos. Debido a esta electrónica las tareas más pesadas y repetitivas podían ser realizadas mediante maquinaria que era controlada por seres humanos, abaratando costes, y aumentando la productividad y la seguridad de los procesos de fabricación.

A finales del siglo XX y a demanda de General Motors se creó el primer PLC (Programmable Logic Controller), dado que se querían sustituir los sistemas que empleaban relés lógicos. La empresa que creó este primer PLC, Bedford Associates, acabó tras una serie de compras y ventas en manos de Schneider Electric, quien es hoy en día una de las mayores empresas de fabricación de componentes electrónicos.⁷

La labor del PLC es la de controlar mediante todo tipo de sensores las funciones que puede realizar una máquina, así como observar la producción de la misma y detectar cualquier fallo en dicha máquina. Otra función que poseen los PLCs es la de comunicarse mediante diferentes métodos (como la red CAN vista anteriormente) entre ellos,

intercambiando datos de los procesos que controla cada PLC y estableciendo un sistema piramidal en cuanto a ordenes se refiere. De modo que dentro de una línea de producción hay un PLC “maestro” que controla y que ejecuta distintos comandos según la información que le transfieren los PLCs “esclavos”. Además hay que mencionar que estos PLCs son altamente modulables, pudiendo añadirle mediante módulos, más memoria, puertos de entrada/salida y adaptadores de red.

Llegados al siglo XXI vemos que los autómatas programables no han sido sustituidos por los PCs de sobremesa, que son algo más baratos y básicamente pueden efectuar las mismas que los PLCs dado que ambos poseen una CPU. Esto se debe a que un PLC está diseñado para funcionar en condiciones extremas de humedad, temperaturas, vibraciones, y perturbaciones eléctricas y magnéticas. También se debe a que los sistemas operativos de los ordenadores son menos estables y fiables que los de un PLC. Todo esto lo hace idóneo para trabajar en un entorno hostil como puede ser una fábrica.

4.2. Elección del autómata

Son muchas las empresas que fabrican PLCs hoy en día, siendo las que más renombre Siemens, Omron, o Schneider por ejemplo. Nos tendremos que decantar por alguno de estos fabricantes en función de varios factores con tal de determinar que autómata industrial vamos a emplear en este Trabajo Final de Grado.

Hemos escogido un autómata por cada una de las marcas nombradas anteriormente.

De la marca Siemens hemos escogido el modelo Siemens 6ES7212-1HE40-0XB0. Este autómata pertenece a la gama media de Siemens y nos puede ser ampliado mediante módulos de expansión, cuenta con 24 entradas/salidas y para las comunicaciones y programaciones emplea un puerto RS-485, que es suficiente para realizar las tareas que deseamos.



Ilustración 18: Siemens 6ES7212-1HE40-0XB0

Por parte de Schneider Electric tenemos el TM221CE16R. Este es muy parejo al anterior autómata con la diferencia de que tiene 16 entradas/salidas en vez de 24 y que posee también un puerto Serial. Dadas sus características sigue siendo apto para el uso que le queremos dar.



Ilustración 19: Schneider Electric TM221CE16R

Por último tenemos el autómata de la marca Omron, modelo CP1L-L10DT-D. Este posee menos entradas/salidas que sus competidores, un total de 10. Como ya hemos dicho, esto no supondrá ningún inconveniente a la hora de desempeñar su tarea. Una cosa a destacar de este autómata a simple vista es que se puede configurar vía conexión USB mediante un ordenador.



Ilustración 20: Omron CP1L-L10DT-D

Ahora vamos a comparar estos 3 autómatas y determinaremos cual es mejor según su conectividad, software empleado y precio.

Conectividad

En este apartado vamos a determinar las formas que tiene cada autómata de comunicarse con el calculador del habitáculo y con el entorno. También valoraremos qué tipo de alimentación tiene cada autómata.

Modelo	Puerto RJ-45	Puerto USB	Conexión CAN	Tipo de alimentación
Siemens 6ES7212-1HE40-0XB0	Si	No	No	24V DC
Schneider TM221CE16R	Si	No	No	230V AC
Omron CP1L-L10DT-D	Si	Si	No	24V DC

Tabla 2: Características de los autómatas programables

A simple vista podemos apreciar que ninguno de los autómatas listados posee conexión CAN. Esto sucede con la inmensa mayoría de autómatas del mercado. La solución que existe a este problema es la de instalar un módulo de expansión para que el autómata reconozca redes CAN o emplear un convertidor CAN/PROFINET.

Por otra parte podemos ver que el modelo de Schneider requiere de una alimentación en corriente alterna de 230 Voltios, que es el estándar que se gasta en todo el mundo, pero que en un vehículo ese tipo de alimentación no existe. En los vehículos actuales los

bornes de la batería funcionan a 12 Voltios en corriente continua, siendo mucho más similar a los 24 Voltios en corriente continua que requieren los autómatas de los fabricantes Siemens y Omron. Podremos pasar los 12 V de la batería a 24 V fácilmente a través de un pequeño transformador que podemos adquirir hoy en día en cualquier tienda especializada.

Otro de los factores que podemos tener en cuenta es el empleo de un puerto USB para la transferencia de los programas al autómata, siendo el autómata del fabricante Omron el único que posee uno. Aunque esta característica sea útil, no es relevante a la hora de seleccionar el autómata con el que vamos a trabajar.

Como conclusión al apartado de la conectividad podemos decir que la mejor opción es el autómata del fabricante Omron, seguido muy de cerca por el del fabricante Siemens

Software empleado

En este apartado vamos a ver qué programas emplea cada fabricante para la programación del autómata, así como las compatibilidades con otros programas o herramientas que nos aporta cada fabricante.

Schneider emplea para todos sus autómatas el software SoMachine, que según nos indica en la web del fabricante es muy sencillo, intuitivo y además permite programar HMI y dispositivos remotos con el mismo software.

Siemens utiliza el software Tia Portal, con el que ya hemos trabajado en más de una ocasión y podemos asegurar que se adapta al trabajo que queremos realizar. Este también permite programar HMI y otros dispositivos remotos. Otra ventaja que tenemos es que podemos crear programas ejecutables para transferirlos a otros dispositivos, como por ejemplo una tablet, para así controlar el autómata desde esta.

Omron tiene un software denominado CX-Programmer con el cual se pueden programar toda su gama de autómatas. El problema que tiene este programa es que no puede programar HMIs (Human-Machine Interface) ni otros dispositivos externos. Para ello se necesita otro software denominado Vijeo Designer, el cual permite controlar HMIs. Por tanto aunque tengamos mucha experiencia a la hora de programar autómatas Omron, es más engorroso y menos intuitivo que las otras alternativas.

Podemos concluir que el software empleado por Siemens está a la par que el de Schneider, dado que son sencillos, intuitivos y permiten controlar numerosos dispositivos externos. Nos decantaremos finalmente por el fabricante Siemens dado que ya hemos trabajado con él y conocemos de antemano sus características y funciones.

Precio

Otro de los aspectos a tener en cuenta a la hora de decantarnos por uno de los autómatas es el precio de cada uno de estos. Hemos buscado en varios distribuidores on-line y las mejores páginas para la compra de autómatas son entre otras es.rs-online.com y www.contaval-online.com. De estas webs hemos sacado el mejor precio de cada autómata antes expuesto.

RS logo | Buscar por palabra clave o nº de ref. | Iniciar sesión / Registrarse | Carrito (0)

Todos los productos | Fabricantes | Novedades | Mi cuenta RS | Ser

Página Principal > Automatización y Control de Procesos > Automatas Programables, Interfaz Hombre-Máquina y Adquisición de Datos > CPUs para Automatas Programables

CPU para PLC Omron CP1L, Salida Transistor, Memoria 10K Words, Puerto USB periférico, Programa 5K steps

Código RS **760-9137**
Fabricante Omron
Nº ref. fabric. **CP1L-L10DT-D**

OMRON

1 Disponible para entrega en 2 día(s) laborable(s).

Precio con IVA
235,54 €
Precio Unidad

unidades | Por unidad

Ilustración 21: Precio del autómatas de Omron

RS logo | Buscar por palabra clave o nº de ref. | Iniciar sesión / Registrarse | Carrito (0)

Todos los productos | Fabricantes | Novedades | Mi cuenta RS | Ser

Página Principal > Automatización y Control de Procesos > Automatas Programables, Interfaz Hombre-Máquina y Adquisición de Datos > CPUs para Automatas Programables

CPU para PLC Schneider Electric Modicon M221, Salida Digital, Programa 10.000 pasos, 16 Puertos E/S

Código RS **806-6837**
Fabricante Schneider Electric
Nº ref. fabric. **TM221CE16R**

Schneider Electric

3 Disponible para entrega en 24/48 horas

171,90 €
Precio Unidad

unidades | Por unidad

Ilustración 22: Precio del autómatas de Schneider

contaval online

ENTREGA EN 24h | GARANTÍA DEVOLUCIÓN | PAGOS SEGUROS | CARRO DE LA COMPRA

Gratis a partir de 94,99€ | en un plazo de 21 días | tu compra sin riesgos | vacío

Bienvenid@ a tu tienda online de automatización

PRODUCTOS | FABRICANTES | CONSULTA A UN TÉCNICO | Oportunidades | OUTLET

6ES7212-1HE40-0XB0 - st70-1200-simatic s7 1200

BUSCAR

Ayuda técnica a profesionales y particulares

Pagos Seguros

6ES7212-1HE40-0XB0 - st70-1200-simatic s7 1200

Referencia: 6ES7212-1HE40-0XB0 | Condición: Nuevo

24-48h

Simatic s7-1200, cpu 1212c, cpu compacta, dc/dc/relés, e/s integradas: 8 ed 24v dc; 6 sd relés 2a; 2 al 0 - 10v dc, alimentación: dc 20,4 - 28,8 v dc, memoria de programa/datos 50 kb

212,28 € -22%

Ilustración 23: Precio del autómatas de Siemens

Como vemos en las ilustraciones el autómatas del fabricante Schneider es mucho más asequible que el de los otros dos fabricantes, ya que lo podemos adquirir por 171,90 €. Le sigue de lejos el autómatas de Siemens por 212,80 € y queda muy atrás el del fabricante Omron con un precio de 235,54 €.

Conclusión

Para poder decantarnos por uno de los autómatas, vamos a compararlos mediante la siguiente tabla, en la que mostraremos las características más relevantes que los diferencian.

Modelo	Software empleado	Puerto USB	Tipo de alimentación	Precio (€)
Siemens 6ES7212-1HE40-0XB0	Tia Portal	No	24V DC	212,28
Schneider TM221CE16R	So Machine	No	230V AC	171,9
Omron CP1L-L10DT-D	CX-Programmer	Si	24V DC	235,54

Tabla 3: Comparativa de los autómatas programables

Finalmente nos hemos decantado por el autómata del fabricante Siemens modelo 6ES7212-1HE40-0XB0 por las siguientes razones:

-Es alimentado mediante 24 V en corriente continua, fácil de conseguir mediante la batería de 12V en corriente continua y un pequeño transformador.

-No tiene el precio más bajo pero a su vez está contenido dentro de su gama.

-Emplea Tia Portal que es un software de programación muy intuitivo, que permite programar con el mismo programa HMIs e incluso dispositivos remotos. Nos da la posibilidad de crear archivos ejecutables y transferirlos a otros PCs o tablets.

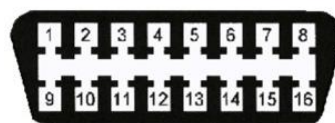
-Tenemos experiencia a la hora de programar mediante el software de Siemens.

A partir de ahora explicaremos los procedimientos a seguir teniendo en cuenta la elección que hemos tomado en este apartado.

4.3. Conexión e instalación del autómata

Ahora que ya hemos escogido el autómata, tenemos que hacer una serie de cosas antes de su instalación y programación.

Primero de todo tenemos que establecer la comunicación entre el autómata y el controlador del habitáculo del Fiat Stilo. Esta comunicación se hará a través del puerto EOBD, que está ubicado en la parte inferior del panel de instrumentos.



1 – Sin uso	9 – Sin uso
2 - J1850 Bus positivo	10 - J1850 Bus negativo
3 – Sin uso	11 – Sin uso
4 - Tierra del Vehículo	12 – Sin uso
5 – Tierra de la Señal	13 – Tierra de la señal
6 - CAN High	14 - CAN Low
7 - ISO 9141-2 - Línea K	15 - ISO 9141-2 - Línea L
8 – Sin uso	16 - Batería - positivo

Ilustración 24: Pines del conector EOBD

Volvemos a recalcar esta imagen dado que es muy importante para todo este apartado.

Mediante un conector o macho de EOBD conseguiremos tener acceso a los pines 6-CAN High y 14-CAN Low. Podemos conseguir un conector de este tipo muy fácilmente a través de internet. Sin ir más lejos tenemos este de www.amazon.es por 2,83 €:



MagiDeal OBD 2 Conector 16 Pines
Adaptador Conectores Macho Herramienta
para Diagnóstico

de Magideal
Sé el primero en opinar sobre este producto

Precio: EUR 5,09
Precio actual: **EUR 2,83** Envío GRATIS.
Ahorras: EUR 2,26 (44%)
Precio final del producto

Ilustración 25: Conector macho puerto EOBD



Ahora que ya tenemos un fácil acceso a la red CAN del vehículo, tenemos que conectar esta red al autómata Siemens. Dado que este autómata no tiene de serie un conector CAN, habrá que adquirir un módulo de expansión o un convertidor que si tenga este conector para poder realizar dicha conexión.

Buscamos un módulo de expansión Siemens que sea compatible con nuestro autómata y que nos permita conectar redes CAN a este. El modelo que buscamos es concretamente el 6GK7243-5DX30-0XE0. Hemos visitado varios distribuidores on-line y visitado la página oficial de Siemens, pero no logramos obtener el precio de dicho módulo, por lo que optaremos por emplear un convertidor CAN/PROFINET.⁸

Podemos adquirir un convertidor de red CAN a red PROFINET en el distribuidor on-line www.adfweb.com. Gracias a este convertidor conseguiremos comunicar con éxito el controlador del habitáculo y el autómata.

Solicitamos presupuesto para el convertidor HD67606-A1 a dicho distribuidor y nos contestan lo siguiente:

ADFweb.com - Request quotation Recibidos x

 **Daniele Dalla Torre (ADFweb.com)** <d.dallatorre@adfweb.com>
 para mí

Sr Carlos Vicente López ,
 Compagnia UPV Valencia

Yo soy Daniele dalla Torre, asistente de ventas de ADFweb.com,
 Especialistas en convertidores de protocolo

Sobre su solicitud:
n. 1 HD67606-A1 PROFINET / CAN - Converter (Ethernet Switch Inside for enter/exit connection, Housing type: A, Terminal Blocks Connectors)

En lo que sigue, usted encontrará, los precios de lista y descuentos reservado para usted

Ofrecemos los siguientes descuentos
15 % de descuento sobre el precio de lista para una venta directa.

por lo tanto, sus precios son los siguientes:

Gateway PROFINET / CAN / CANopen:
339.15 Euro /Piece for HD67606-A1 - (PROFINET / CAN)
381.65 Euro /Piece for HD67607-A1 - (PROFINET / CANopen)

El costo de envío, está a sólo 19 Euro,
 Los productos están en almacén y pueden ser enviados hoy. El envío es por mensajería, DHL o Fedex, El viaje tarda 1 días en SPAIN,

Ilustración 26: Correo recibido de www.adfweb.com

Cómo podemos observar, dicho convertidor nos saldrá por 339,15 € más 19 € de gastos de envío. Resulta una pieza muy cara, más aún que el propio autómata, pero sin esta pieza la comunicación entre el controlador del habitáculo y el autómata sería imposible.

Una vez hemos comunicado ambos elementos de nuestro sistema mediante un cable Ethernet, procederemos a alimentar el autómata y el convertidor CAN/PROFINET. El puerto EOBD nos facilita esta tarea dado que tiene los pines 16-Batería positivo y 4-Tierra del vehículo. De estos pines podemos extraer 12 V en corriente continua sin mayor problema.

Dado que nuestro autómata y nuestro convertidor necesitan una alimentación de 24 V y el puerto EOBD suministra 12 V, necesitaremos un transformador para pasar de una tensión a la otra. Un transformador de 12 V a 24 V es fácil de conseguir a través de internet. En www.amazon.es tenemos transformadores de este tipo desde los 10,99 €.



Ilustración 27: Transformador de 12 V a 24 V

Una vez hemos conectado los bornes positivo y negativo del autómata y del convertidor con la salida del transformador, procederemos a ubicar todas las piezas físicas que componen nuestro sistema dentro del vehículo. Lo más lógico sería colocar todo lo más cerca posible del cuadro de instrumentos, que es donde está ubicado el puerto EOBD.

Una buena ubicación sería la guantera del vehículo, dado que el Fiat Stilo posee 2 guanteras en la parte del copiloto y no echaríamos en falta espacio. De esta forma tendríamos que pasar un conjunto de cables compuesto por CAN High, CAN Low, positivo de la batería y tierra del vehículo, por el interior del salpicadero hasta la zona donde está ubicado el puerto EOBD.

Otra ubicación posible sería el maletero, dado que hay espacio de sobra y las distancias desde el conector EOBD hasta nuestro sistema no serían muy grandes. Recordemos que la red CAN admite longitudes de cable de hasta 1000 metros sin emplear repetidores. Esta opción nos aportaría más espacio en las guanteras aunque por otro lado si quisiéramos añadir más cableado en un futuro sería complicado pasarlo con facilidad del maletero hasta la zona delantera del habitáculo del vehículo.

Por tanto optaremos por emplazar el transformador, el autómata y el convertidor en la guantera del Fiat Stilo, restándonos espacio de la guantera, pero siendo más fácil la instalación y el cableado del sistema.

4.4. Programación y gestión de datos

Ahora que ya tenemos todos los elementos de nuestro sistema definidos y comunicados físicamente, vamos a ver cómo se haría la programación inicial y cómo podríamos gestionar los datos obtenidos a través del calculador del habitáculo.

Primero necesitamos el software denominado Tia Portal v14 ya que es totalmente compatible con Windows 10. En nuestro caso tenemos la versión Professional, que es la que más posibilidades nos ofrece, lo que nos permitirá programar para una multitud de dispositivos, permitiéndonos incluso crear archivos ejecutables para poder emplearlos en otros dispositivos, como tablets u ordenadores. Cabe destacar que esta versión es de pago, pero la tenemos a nuestra disposición dado que hemos trabajado previamente con el fabricante Siemens y con esta gama de autómatas.

También necesitaremos el software para configurar nuestro convertidor. El programa se denomina SW67607 y lo podemos encontrar en la siguiente dirección web:

<http://www.adfweb.com/home/download/download.asp>

Todo el software de esta página es gratuito y cada programa viene con un manual de usuario donde explica cómo utilizar y configurar el aparato que el software controla.

Configuración de convertidor

Una vez instalados los componentes y software necesarios, vamos a ver cómo se haría la configuración inicial del convertidor mediante el programa SW6706.

Primero de todo crearemos y le daremos nombre a la nueva configuración haciendo click en “New Configuration” del menú principal del programa.

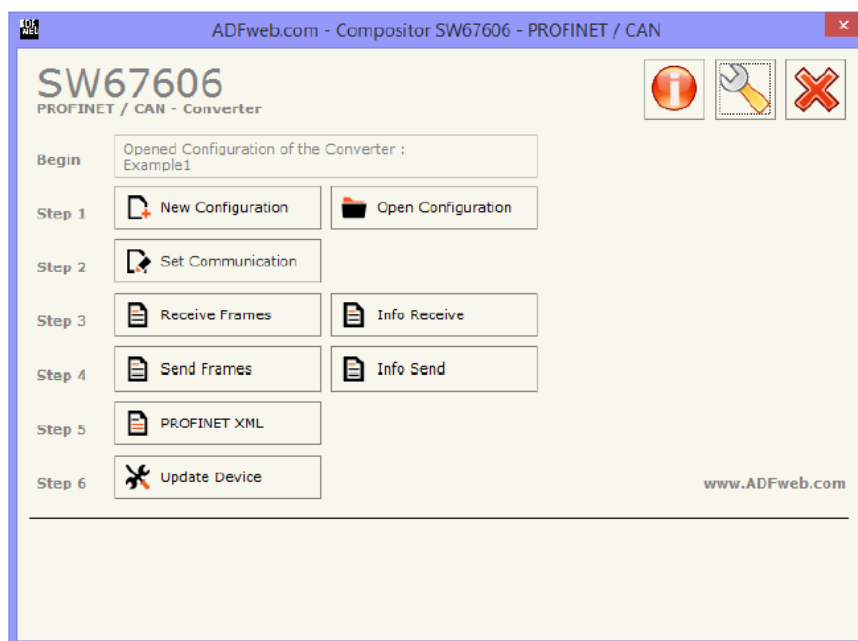


Ilustración 28: Menú principal del SW67606

Después nos iremos al segundo paso haciendo click en “Set Communication”. Aquí designaremos y configuraremos la dirección IP del convertidor en la que será la red PROFINET. También introduciremos el baudrate con el que trabaja la red CAN del vehículo.

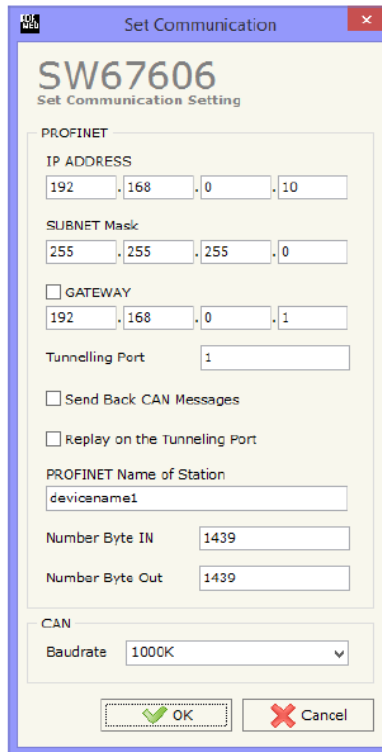


Ilustración 29: Menú "Set Communication" del SW67606

En el tercer paso haremos click en "Recive Frames" y configuraremos como recibe el convertidor los datos de la red PROFINET.

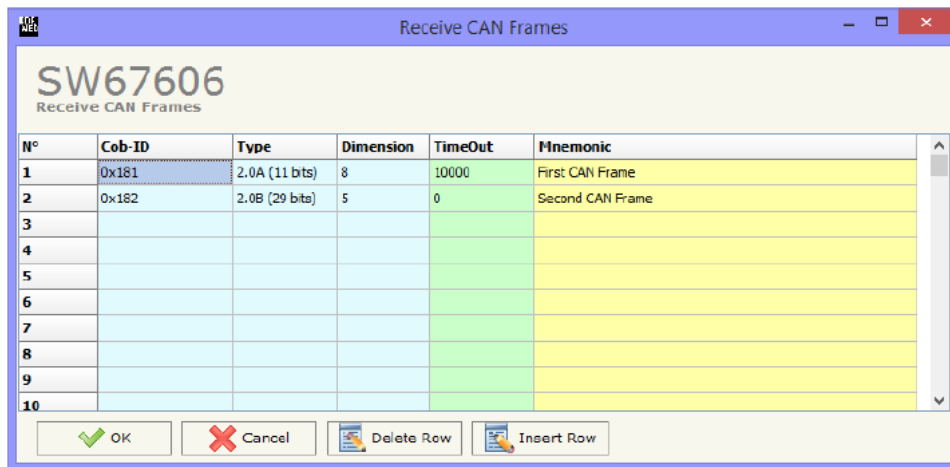


Ilustración 30: Menú "Recieve Frames" del SW67606

Después mediante el menú "Info Recieve" configuraremos que byte de la red CAN corresponde a que byte de la red PROFINET.

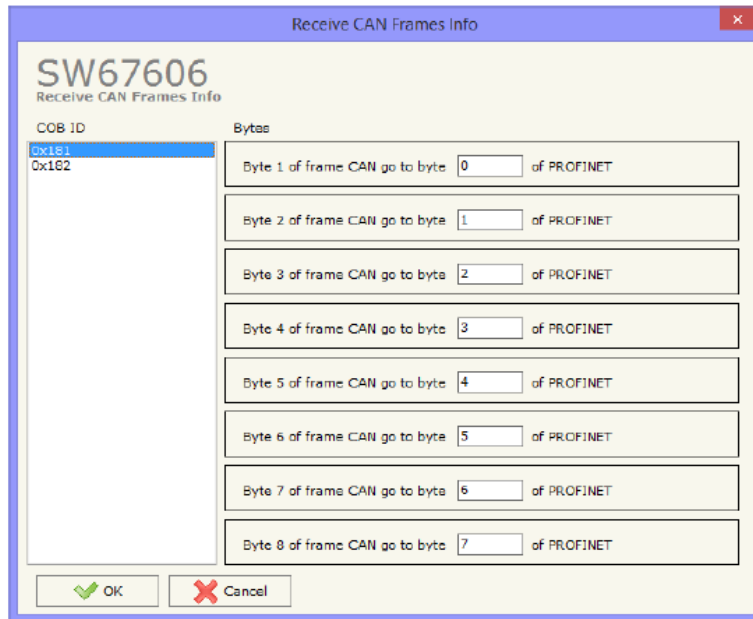


Ilustración 31: Menú "Info Recieve" del SW67606

En el cuarto paso configuraremos como se envía la información desde el convertidor a la red PROFINET. En el menú "Send Frames" introduciremos los datos que se enviarán a la red PROFINET desde el convertidor.

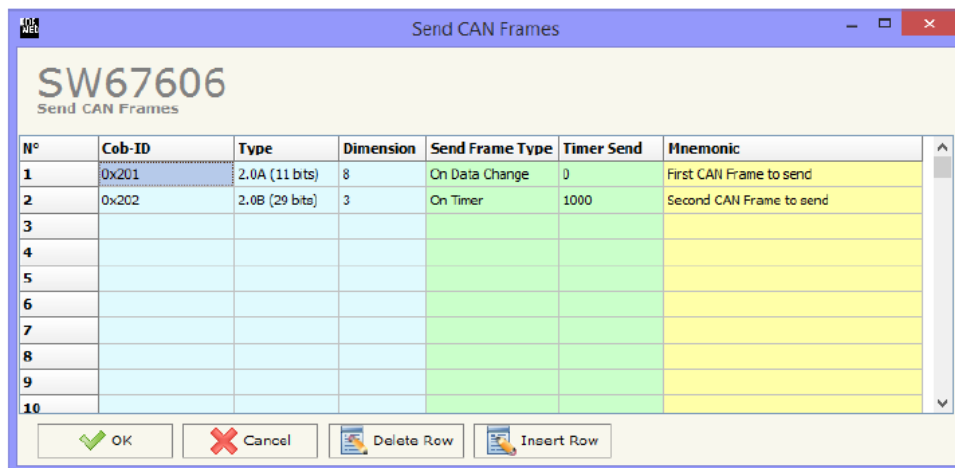


Ilustración 32: Menú "Send Frames" del SW67606

Después mediante el menú "Info Send" configuraremos nuevamente que byte de la red CAN corresponde a que byte de la red PROFINET.

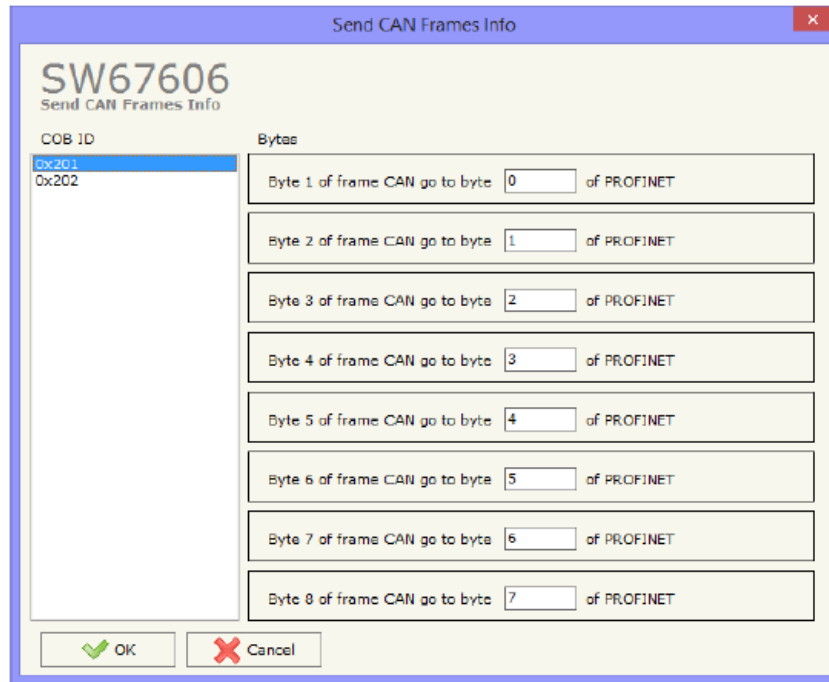


Ilustración 33: Menú "Info Send" del SW67606

Por último generaremos un fichero xml mediante el menú "PROFINET XML" que podrá ser importado al maestro de la red PROFINET. Posteriormente cargaremos el archivo xml en el autómata mediante el programa Tia Portal V14

Configuración del autómata

Ya sabiendo configurar el convertidor, pasaremos a configurar el autómata en sí.

Primero de todo crearemos un nuevo proyecto mediante el programa Tia Portal V14 y nos situaremos en la ventana "Vista del proyecto".

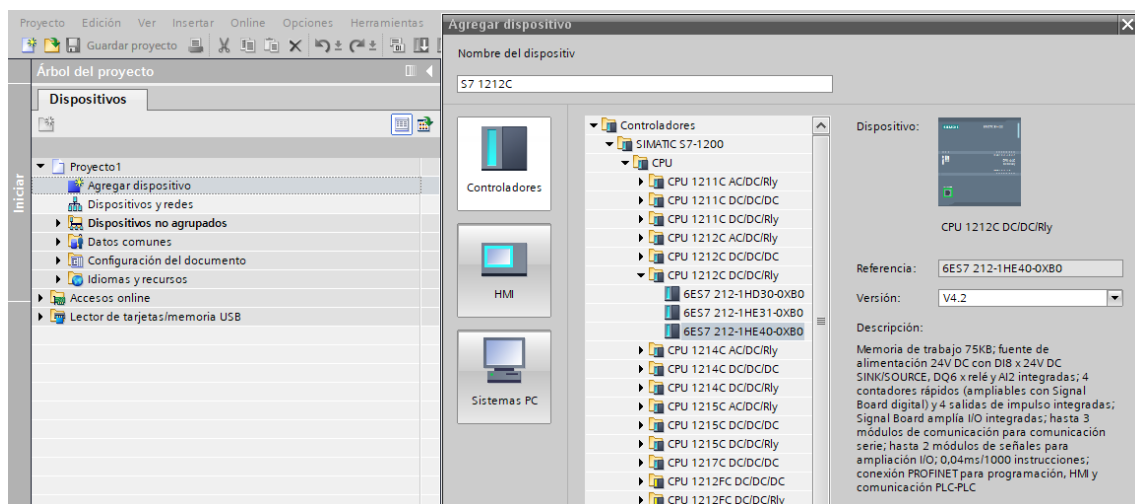


Ilustración 34: Selección de autómata en Tia Portal V14

A continuación haremos doble click en "Agregar dispositivo" y nos aparecerá una pestaña donde podremos seleccionar que dispositivo queremos añadir. En concreto nos iremos a Controladores → SIMATIC S7-1200 → CPU 1212C DC/DC/Rly. Aquí seleccionaremos el modelo de nuestro autómata que es el 6ES7212-1HE40-0XB0 y le daremos a aceptar.

Hecho esto, habrá que añadir ahora el convertidor para que el Tia Portal lo reconozca. Para ello hay que localizar el fichero yendo a Opciones → Administrar archivos de descripción de dispositivos.

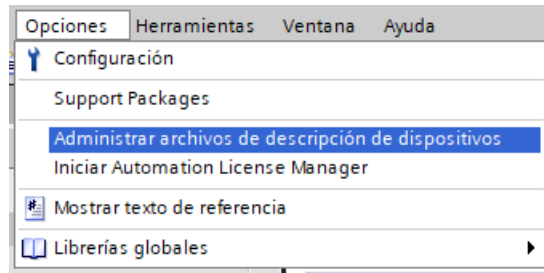


Ilustración 35: Búsqueda de archivo xml mediante Tia Portal V14

Nos aparecerá una ventana donde introduciremos la ubicación de la carpeta donde se encuentra el archivo xml. Una vez seleccionado el archivo procederemos a instalarlo.

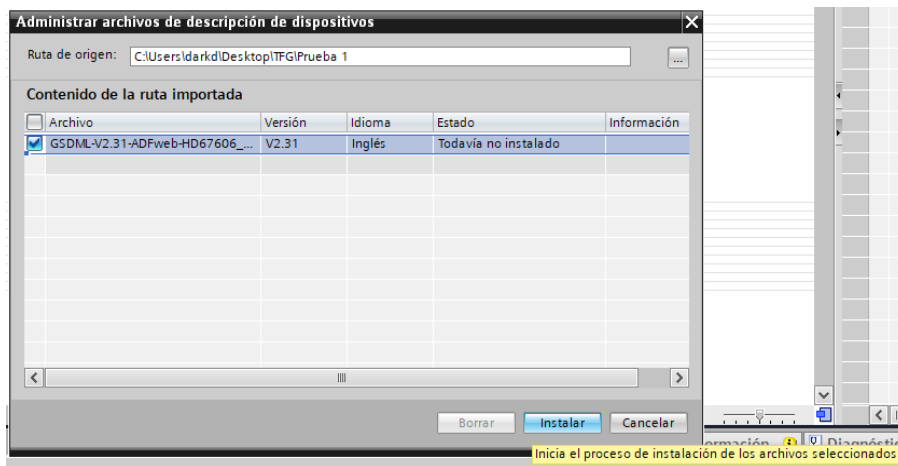


Ilustración 36: Instalación de archivo xml en Tia Portal V14

Una vez instalado habrá que buscarlo en el catálogo del Tia Portal. Nos iremos a Otros dispositivos de campo → PROFINET IO → Gateway → ADFweb.com → HD67606 y haremos doble click en “CAN” de forma que añadiremos el convertidor a nuestro programa.

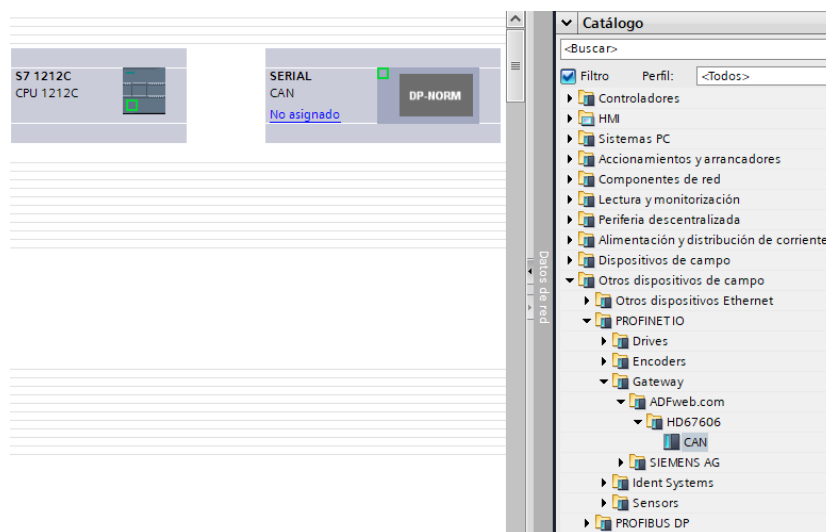


Ilustración 37: Búsqueda de convertidor en el Tia Portal V14

Ahora procederemos a enlazar virtualmente el autómatas y el convertidor. Para ello nos situamos en la pestaña de “Dispositivos y redes” y pinchando en el recuadro verde de la imagen del autómatas, arrastramos hasta el recuadro verde del convertidor.

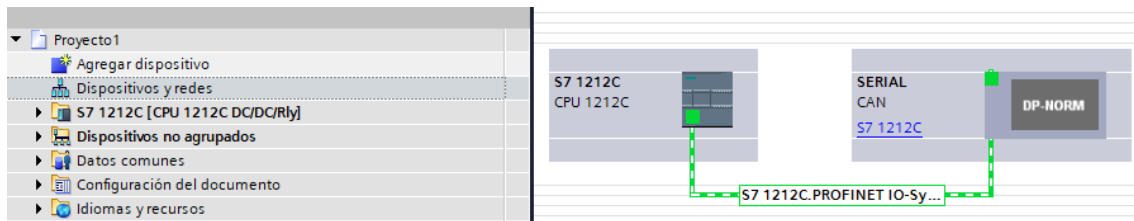


Ilustración 38: Conexión entre autómatas y convertidor en Tia Portal V14

El último paso que deberemos hacer será el de asignar un nombre y una dirección ip al convertidor. Haremos doble click en el convertidor y modificaremos los valores que antes hemos descrito desde las pestañas “General” y “Direcciones Ethernet” que se encuentran en el apartado “Interfaz PROFINET [X1]” de la pestaña “General”.

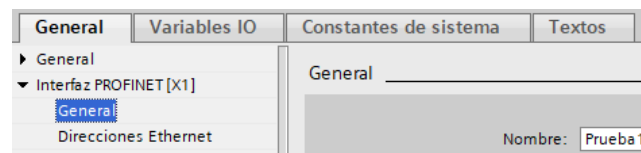


Ilustración 39: Asignar nombre al convertidor mediante Tia Portal V14

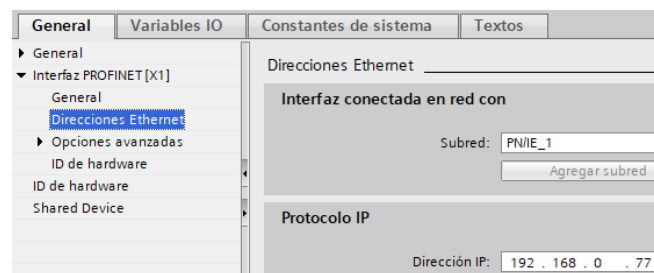


Ilustración 40: Asignar dirección ip mediante Tia Portal V14

Una vez realizados todos los pasos podremos leer la información a través de la red CAN y por tanto asignar variables y estados para cada variable. Solamente queda concretar de qué forma enviaremos la información de las variables y su estado a una pantalla, para su visionado desde el habitáculo del Fiat Stilo.

4.5. Visualización de datos a través de pantalla

Habiendo explicado cómo se configura inicialmente el Tia Portal V14 y el convertidor para poder trabajar con las variables que queramos de la red CAN del Fiat Stilo, vamos a ver cómo podemos transmitir toda esta información a una pantalla que esté ubicada en la zona central del salpicadero y también veremos que pantalla se adapta mejor a esta alternativa.

Ubicación de pantalla

Dado que contamos con una zona tan delimitada como es la zona central del salpicadero del vehículo, necesitamos las medidas de esta zona para determinar el tamaño máximo de nuestra pantalla.



Ilustración 41: Zona central del salpicadero del Fiat Stilo

Las medidas de la zona central del salpicadero son de 260 mm de largo por 165 mm de alto. La pantalla no necesita ser del tamaño máximo que disponemos en el salpicadero, ya que existen una gran cantidad de embellecedores para que el elemento que tengamos instalado en esta zona quede encuadrado y con un buen acabado.

Selección de pantalla

Ya sabiendo el tamaño máximo que puede ser nuestra pantalla, vamos a ver cuál sería el mejor display para visualizar toda la información. Las dos opciones que nos ofrece el Tia Portal son la de un HMI (Human Machine Display) o la de un dispositivo capaz de ejecutar archivos compatibles con Tia Portal, como puede ser una tablet con Windows 10. Cabe destacar que requiere el mismo esfuerzo el trabajar con un HMI o una tablet, así que nos basaremos en las especificaciones técnicas de cada dispositivo a la hora de escoger uno de los dos.

En busca de un display HMI que se ajuste a nuestras necesidades hemos encontrado en el distribuidor online www.contaval-online.com el modelo 6AV2124-0GC01-0AX0. Este tiene una pantalla de 7", con una resolución de 800 x 480 píxeles, pantalla TFT y táctil, unas dimensiones de 214 x 158 x 63 mm y un precio de 840 €.



6AV2124-0GC01-0AX0 - st801 panel-simatic hmi paneles

Reference: 6AV2124-0GC01-0AX0 | Condición: Nuevo | 6 artículos

Entrega inmediata

Simatic hmi tp700 comfort panel, windows ce 6.0, display tft panorámico de 7", 12 mb de memoria de config., configurable con wincc v11 o sup.¡atención!: precio oferta. precio neto!

840,00 €

1

Ilustración 42: Modelo de HMI escogido

Por otra parte tenemos la tablet modelo Teclast X80 Pro que podemos encontrar en www.amazon.es. Esta tiene una pantalla de 8", con una resolución de 1920 x 1200 píxeles, pantalla IPS y táctil, unas dimensiones de 209 x 126 x 10 mm y un precio de 72,99 €.



Ilustración 43: Modelo de tablet escogido

Vistas las características de ambos sistemas y recalando que no existe mucha diferencia a la hora de programar y trabajar con ambos, la elección es obvia. La tablet es mejor en todos los aspectos que hemos nombrado, incluyendo el precio dado que es 767 € más barata. También hay que mencionar que la tablet es un mini-pc por lo que nos puede servir de centro multimedia en un futuro, cosa que el HMI no puede hacer ya que únicamente es una pantalla.

La razón por la que el HMI es tan caro es debido a que está preparado para trabajar bajo situaciones extremas típicas en las fábricas. Y dado que en el habitáculo de un vehículo no se dan este tipo de situaciones, estas protecciones son innecesarias.

Instalación de pantalla

Ahora que ya hemos seleccionado la tablet, esta tendrá que alimentarse a 5 V en corriente continua. Emplearemos la toma de corriente del puerto EOBD nuevamente y con la ayuda de un transformador de 12 V a 5 V alimentaremos la tablet. Hemos escogido el siguiente transformador de www.amazon.es ya que posee dos puertos usb en vez de uno y admite carga rápida a un precio razonable de 7,01 €.



Ilustración 44: Transformador de 12 V a 5 V

Llegados a este punto nos encontraremos con un pequeño problema, y es que necesitamos conectar la pantalla vía Ethernet al autómata, pero este ya está conectado al convertidor. La solución pasa por emplear un switch de puertos Ethernet, que nos permitirá interconectar a una misma red varios dispositivos. Podemos encontrar el modelo Tp-link tl-sf1005d, que cumple todos nuestros requisitos, en www.amazon.es por 10,16 €. Este tendrá que alimentarse a 5 V en corriente continua, por lo que podremos emplear el transformador que ya tenemos instalado para la tablet para alimentar también el switch.



TP-LINK TL-SF1005D - Switch de red con 5 puertos 10/100 Mbps (IEEE 802.3x, sin configuración) blanco

de TP-Link
 ★★★★★ 782 opiniones de clientes
 | 68 preguntas respondidas

Precio recomendado: EUR 12,90
 Precio: **EUR 10,16** Envío GRATIS en pedidos superiores a 29€. Ver detalles
 Ahorros: EUR 2,74 (21%)
 Precio final del producto

Ilustración 45: Switch de puertos Ethernet

Necesitamos también un adaptador Ethernet a micro-usb para poder conectar la tablet al switch de puertos Ethernet mencionado anteriormente. De este modo la tablet ya estará conectada con el resto del sistema. Utilizaremos un simple adaptador de www.amazon.es para esta labor. Hemos escogido el siguiente por un precio de 9,99 €.



Aukru® Micro USB a Ethernet 10/100 Mbps RJ45 de red adaptador LAN para Android Tablet

de Aukru
 ★★★★★ 17 opiniones de clientes
 | 8 preguntas respondidas

Precio: EUR 12,99
 Precio actual: **EUR 9,99** Envío GRATIS en pedidos superiores a 29€. Ver detalles
 Ahorros: EUR 3,00 (23%)
 Precio final del producto

Ilustración 46: Adaptador Ethernet a micro-usb

Ya que la tablet cuenta únicamente con un puerto micro-usb, necesitaremos un ladrón de puertos micro-usb para poder cargar la tablet mediante uno de los puertos y para poder realizar la conexión a la red mediante el otro puerto. El ladrón de puertos que hemos escogido lo podemos encontrar en www.amazon.es por un precio de 8,55 €.



XCSOURCE® Dual Doble Micro USB Host Anfitrión OTG Hub Cable Adaptador para Nota2 S3 S4 Nexus4 5 HTC Uno AC155

de XCSOURCE
 ★★★★★ 69 opiniones de clientes
 | 18 preguntas respondidas

Precio: **EUR 8,55** Envío GRATIS en pedidos superiores a 29€. Ver detalles
 Precio final del producto

Promoción disponible 8 promociones

Ilustración 47: Ladrón de puertos micro-usb

Para finalizar ubicaremos la Tablet en la parte media del salpicadero, eliminando la Radio/CD que ya existe dado que en un futuro se podría sacar el sonido directamente de la tablet mediante amplificadores. Ya que hemos escogido la Tablet según el tamaño del hueco del salpicadero, no habrá problemas a la hora de acoplarla en esta ubicación.

Configuración inicial de la pantalla

Por último vamos a establecer la comunicación entre el autómatas y la tablet mediante el Tia Portal V14.

Primero de todo debemos de tener instalado en la tablet el programa Win CC Runtime Loader. Para ello instalaremos previamente el Win CC Professional V14. Hay que recalcar

que no necesitaremos instalar todo el Tia Portal V14 como tenemos en el ordenador, siendo mucho más liviano el programa. De esta forma podremos iniciar el ejecutable que creamos a través del Tia Portal V14 del ordenador en la tablet sin mayor dificultad.

Comenzaremos añadiendo al Tia Portal V14 la tablet. Para ello hacemos doble click en “Agregar dispositivo, y en el menú que nos aparece vamos Sistema PC → SIMATIC HMI Application → WinCC RT Advanced y haremos click en aceptar.

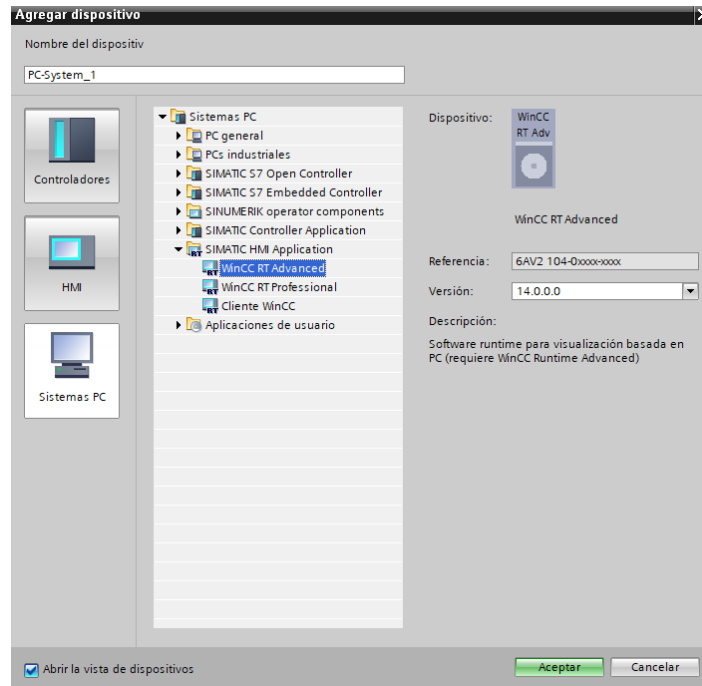


Ilustración 48: Insertar tablet en el Tia Portal V14

A continuación se creará una “Estación PC SIMATIC” que en realidad será la tablet. Tendremos que añadirle un módulo de comunicaciones Ethernet a esta estación para enlazar la red creada anteriormente con la tablet. A este módulo de comunicaciones accederemos desde Catálogo → Módulos de comunicación → PROFINET/Ethernet → IE general. Pincharemos y arrastraremos desde “IE general” hasta la tablet. De esta forma ya tendremos un puerto de red en la tablet, lo que nos permitirá comunicar la tablet con el resto del sistema.

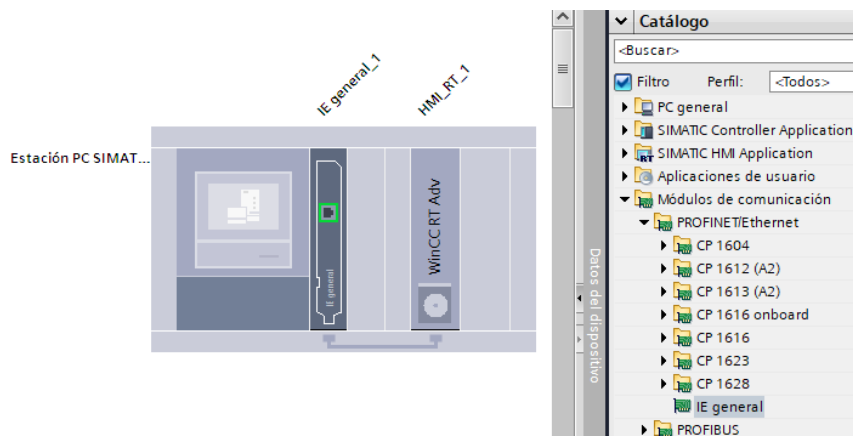


Ilustración 49: Creación de puerto de red para la tablet mediante Tia Portal V14

Para finalizar la conexión entre la tablet y la red, deberemos efectuar los mismos pasos que hicimos para enlazar el autómata con el convertidor. Pincharemos y arrastraremos desde el recuadro verde del autómata hasta el recuadro verde de la tablet. Esto generará otra conexión y una dirección ip para la tablet. Habrá que revisar que las direcciones ip de los 3 elementos de nuestra red no sean iguales.

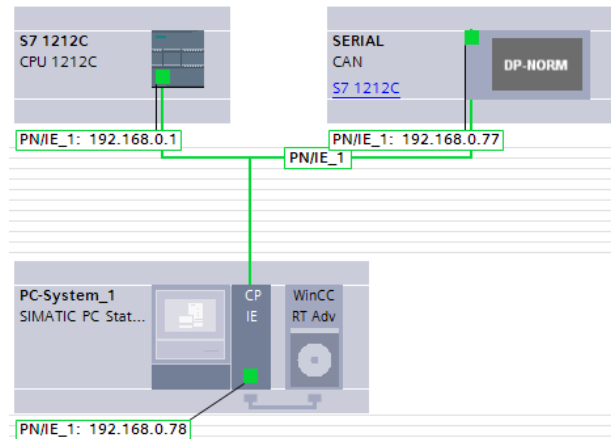


Ilustración 50: Esquema y direcciones ip que componen la red

Por último tendremos que transferir el archivo ejecutable desde nuestro ordenador con Tia Portal V14 hasta la tablet con Win CC Professional V14.

En el programa Tia Portal V14 del ordenador hacemos click en “Cargar en dispositivo” y dejamos la ventana que nos aparezca abierta.

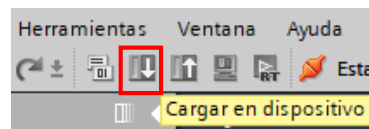


Ilustración 51: Acción "Cargar en dispositivo " en Tia Portal V14

Una vez hecho esto en el ordenador, iniciaremos desde la tablet el programa “Win CC Runtime Loader” y haremos click en “Transferencia” Esto transferirá el archivo ejecutable desde el ordenador hasta la tablet.



Ilustración 52: Menú principal del programa Win CC Runtime Loader

Ahora que ya tenemos el archivo ejecutable en nuestra tablet, podremos ejecutarlo haciendo doble click sobre este, sin necesidad de abrir el Tia Portal V14 o el Win CC Runtime Loader.

Para que funcione todo correctamente simplemente necesitaremos que tanto el autómata, como el convertidor y la tablet estén comunicados entre sí.

Visualización de variables por pantalla

Para finalizar toda la instalación, queda ver cómo ajustar la pantalla y mostrar las variables en la tablet mediante el Tia Portal V14.

Primero de todo vamos a agregar una imagen y una plantilla siguiendo la ruta que se muestra a continuación:

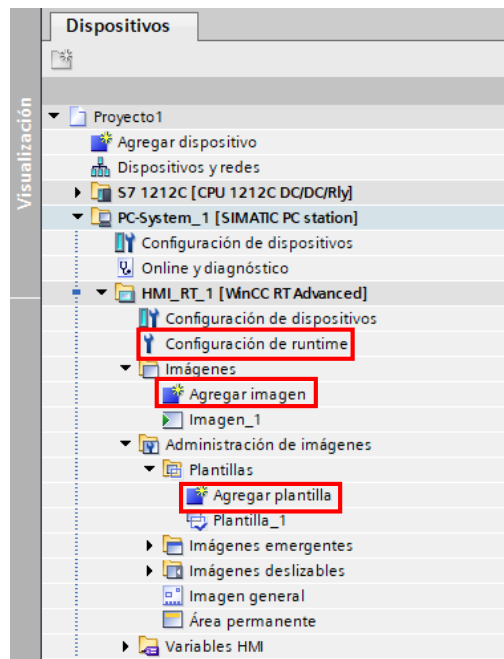


Ilustración 53: Agregar imágenes y plantillas mediante Tia Portal V14

Una vez hecho esto, pasaremos a indicar cuál es nuestra resolución de la pantalla en píxeles, que en nuestro caso es de 1920 x 1200. Para poder configurar este parámetro tendremos que hacer doble click en “Configuración de runtime” tal y como podemos ver en la imagen superior. Al realizar esta acción nos aparecerá un menú donde configurar entre otras opciones la resolución de la pantalla, la cual pondremos a 1920 x 1200.

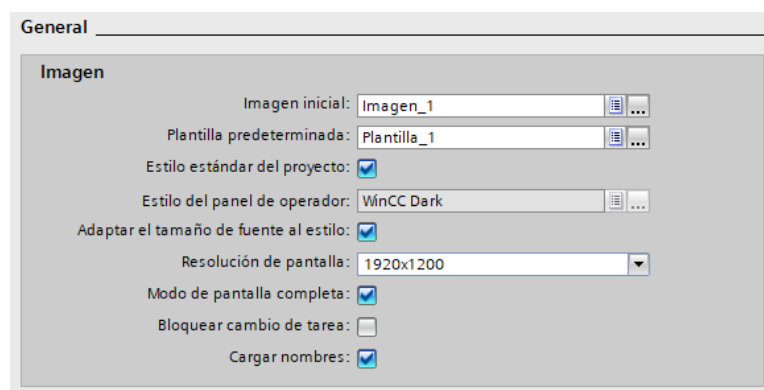


Ilustración 54: Escoger resolución de la pantalla del Tia Portal V14

Por último para visualizar todos los elementos que queramos mediante la pantalla, deberemos de hacer doble click en la imagen que ya tenemos creada y comenzaremos a insertar todos los elementos que queramos, ya sean objetos básicos (líneas, círculos,...) o elementos (velocímetros, pulsadores,...).

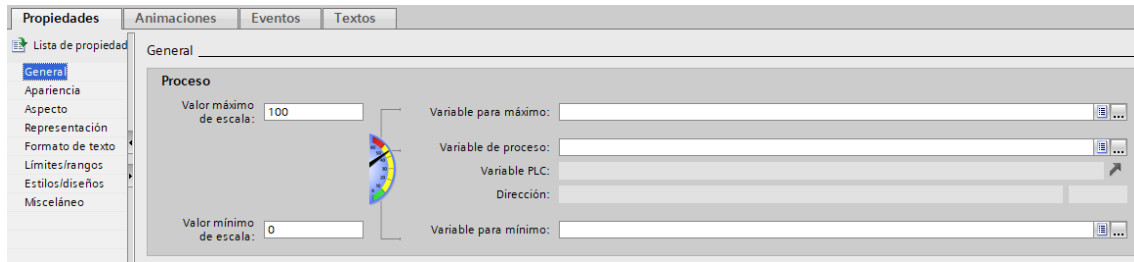


Ilustración 55: Opciones de personalización de un indicador gauge

Cómo podemos ver en la imagen las opciones de personalización de este “indicador gauge” son casi infinitas, al igual que la del resto de elementos que podemos añadir. Además el programa es muy intuitivo a la hora de configurar cada elemento o añadir variables a dichos elementos, ahorrándonos de esta forma mucho tiempo en la elaboración de las pantallas.

4.6. Costes y conclusiones

Habiendo explicado nuestra instalación y las configuraciones iniciales de los 3 elementos que componen todo el sistema, vamos a ver los costes que conlleva realizar este proyecto mediante un autómatas.

En la siguiente tabla mostraremos los elementos que hemos empleado para la elaboración de nuestro sistema, así como su coste. Cabe destacar que los precios de todos los productos pueden variar en función del tiempo. Los precios de los componentes que mostramos a continuación datan de Diciembre de 2017.

Componente	Distribuidor	Precio (€)
Siemens 6ES7212-1HE40-0XB0	Contaval-online	212,8
Conector Macho EOBD	Amazon	2,83
Transformador 12 V a 24 V	Amazon	10,99
Convertidor HD67606-A1	ADFweb	358,15
Tablet Teclast X80 Pro	Amazon	72,99
Transformador 12 V a 5 V	Amazon	7,01
Adaptador Micro-usb a Ethernet	Amazon	9,99
Ladrón de puertos Micro-usb	Amazon	8,55
Switch Tp-link tl-sf1005d	Amazon	10,16
Precio Total (€)		693,47

Tabla 4: Costes de la primera alternativa, autómatas industrial

Vamos a valorar y a sacar conclusiones de los distintos factores relevantes que afectan a la elaboración de todo este sistema a la hora de mostrar la información que deseamos por pantalla.

-Instalación en vehículo: La instalación es algo engorrosa dado que tenemos que alimentar 3 componentes mediante el mismo conector (puerto EOBD) y además necesitamos dos transformadores para que la tensión de alimentación en los 3 componentes sea la correcta. Luego para la conexión de red entre los componentes

necesitamos un ladrón de puertos dado que el autómata no posee más de un puerto RJ45 o Ethernet. También hay que recalcar que necesitamos reducir el espacio de la guantera con tal de que quepan los transformadores, el autómata y el convertidor. A pesar de que la instalación sea algo complicada en realidad es bastante simple, pues no requiere de conocimientos eléctricos o de comunicaciones avanzados.

-Seguridad: La seguridad es un aspecto muy importante en este TFG dado que trabajamos con información importante del vehículo, por lo que un fallo en una lectura o escritura de esta puede ser muy perjudicial tanto para el vehículo como para el sistema instalado. Los autómatas están preparados físicamente para trabajar en entornos donde las temperaturas y perturbaciones son extremas, también manejan un software muy básico para evitar así cualquier cuelgue del sistema y hacer el programa más estable. Todo esto hace idóneo el autómata para el uso que le queremos dar.

-Configuración inicial: Antes de comenzar a diseñar el programa que va a funcionar en el vehículo hay que configurar todos los componentes del sistema para que trabajen correctamente. El convertidor nos trae su propio manual para la configuración, lo que facilita enormemente su instalación. En lo que respecta al autómata y la pantalla, ambos son fáciles de configurar gracias a lo intuitivo que resulta Tia Portal V14.

-Programación: Una vez podemos obtener las variables del vehículo y está todo configurado, programar con el autómata es muy sencillo.

-Visualización por pantalla: Ya que el Tia Portal V14 nos ofrece una librería donde nos aporta muchísimas opciones gráficas y dado que es muy intuitivo, configurar la pantalla para que nos muestre la información que deseemos es muy sencillo.

-Coste en tiempo: De todo este sistema lo que más tiempo requiere es su instalación, debido a la cantidad de cableado que se requiere. Por otro lado el Tia Portal V14 aporta muchas facilidades a la hora de diseñar la pantalla, lo que acorta el tiempo necesario a la hora de finalizar todo el proyecto.

-Coste: En cuestión de precios, el elaborar este proyecto mediante autómata es excesivamente caro dado que se requiere un convertidor CAN/PROFINET el cual es muy costoso y un autómata que tampoco es nada barato. Hay que añadir que aunque tengamos a nuestra disposición el software Tia Portal V14, este también supone un elevado coste.

Ahora que ya tenemos las conclusiones de todos los apartados relevantes, vamos a valorarlos de 1 al 10 de cara a futuras comparativas con el resto de sistemas alternativos que vamos a analizar. Para esta comparativa nos basaremos en la siguiente tabla, donde ya se encuentra la valoración de esta alternativa.

	Instalación en vehículo	Seguridad	Configuración inicial	Programación	Visualización por pantalla	Coste en tiempo	Coste	Valoración Final
Valoración	3	10	7	7	9	5	2	43

Tabla 5: Valoración de la alternativa autómata industrial

5. Solución alternativa: Arduino

5.1. Introducción al Arduino

En este apartado vamos a introducir Arduino de cara a entender que es, conocer su historia y características, comprender su funcionamiento y que cosas podemos hacer con él.

Un Arduino es básicamente una placa con circuito impreso donde encontramos puertos digitales, entradas y salidas analógicas, y un microcontrolador que se encarga de gestionar todos estos elementos. Se trata también de un sistema donde tanto su hardware como su software son libres y de código abierto, por lo que la gente puede emplearlos para una infinidad de aplicaciones y encima por un precio bastante reducido.⁴⁵

El primer arduino, por denominarlo de alguna forma, fue creado en el año 2005 por Massimo Banzi, que era por aquel entonces estudiante en el instituto IVRAE de Italia. Este primer proyecto que todavía no tenía nombre oficial tenía como objetivo ser una herramienta para el aprendizaje de los alumnos de dicho instituto, dado que anteriormente no era barato conseguir una placa de microcontroladores.

Este primer prototipo estaba formado por un microcontrolador y unas resistencias conectadas a un circuito eléctrico. Al principio solamente podían conectarse sensores simples, resistencias o leds.

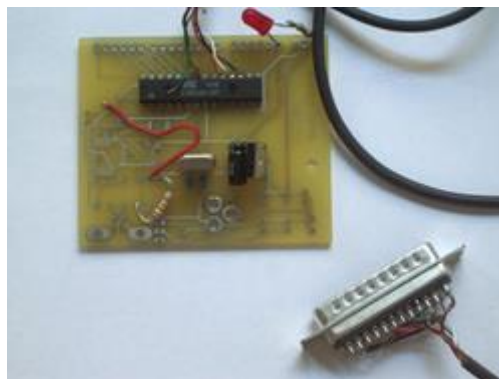


Ilustración 56: Prototipo del primer Arduino

Más tarde se incorporó a este joven proyecto Hernando Barragán, quien contribuyó desarrollando la programación de esta placa. Poco después participaría también David Mellis, actualizando al interfaz del software. Apareció también el estudiante español David Cuartiles, quien ayudó a mejorar el hardware de la placa para así poder dar soporte y proporcionar memoria para almacenar y gestionar el software que controla el sistema. Finalmente se añadió a este equipo Tom Igoe, que tras interesarse por el proyecto fue invitado a unirse a este, lo que conllevó a que se instalara un puerto USB en la placa para conectarla a un ordenador y también a que se comenzara a producir en masa dicha placa para su comercialización.

A partir de aquí se produjeron primero 300 unidades de la placa, que irían destinadas exclusivamente a los alumnos del IVRAE. Por la venta de cada una de estas placas se obtenía una ganancia de 1 \$. Pero se pensó así para que todos los estudiantes de dicho

instituto pudieran tener una y para tratar de salvar la mala situación del IVRAE, que por aquél entonces estaba a punto de cerrar debido a una falta de fondos.

Finalmente Arduino se popularizó dado sus características y se comenzó a vender alrededor de todo el mundo rápidamente.^{9 10}



Ilustración 57: Modelo actual de placa Arduino

A continuación vamos a explicar que componentes tiene una placa de Arduino y que función tiene cada uno de estos. Tomaremos como modelo la placa Arduino UNO, dado que es la más común y estandarizada.¹¹

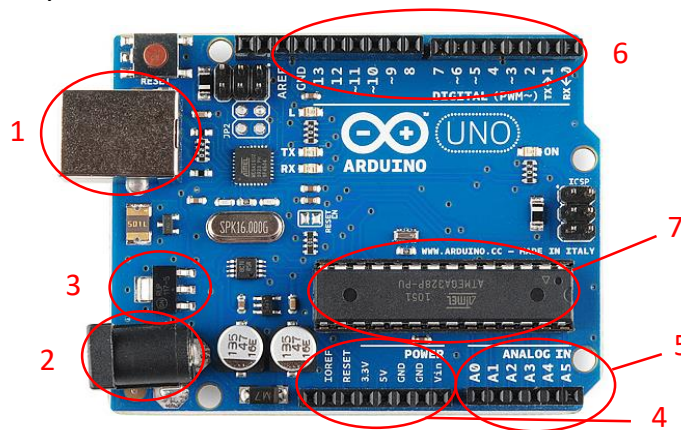


Ilustración 58: Placa Arduino UNO

1- Puerto USB: mediante este puerto el Arduino puede conectarse con un ordenador para así cargar los programas que queramos. Este puerto también puede emplearse para alimentar eléctricamente el Arduino, siendo la tensión necesaria de alimentación 12 V en corriente continua.

2- Toma de corriente: Este puerto está destinado completamente a alimentar el Arduino. Como hemos visto antes, se alimenta a 12 V en corriente continua.

3- Regulador de voltaje: Como su propio nombre indica, es un regulador que se encarga de garantizar que la tensión de alimentación que se le suministra al Arduino es la correcta. Por tanto este nos impedirá alimentar el Arduino a más de 20 V, dejándonos un margen de 6 a 20 Voltios como tensión de alimentación admisible.

4- Mediante estos pines podemos alimentar distintos elementos. Tenemos un pin que suministra 5 V en corriente continua, otro 3,3 V en corriente continua y luego tenemos 2 pines hacen de tierra de los circuitos que implementemos.

5- Estos pines tienen la función de conectar sensores analógicos, como un termómetro o un sensor de proximidad, para convertir la señal que nos transmiten estos sensores en valores digitales.

6- Estos pines sirven tanto para la entrada como para la salida de información en formato digital. Los pines con el símbolo (~) pueden utilizarse también para la modulación por ancho de pulso, que se emplea principalmente para controlar motores eléctricos.

7- Este es el microcontrolador, que es el cerebro de toda la placa. Se encarga de controlar toda la información que recibe o suministra desde los distintos pines de la placa. Es el elemento que almacena los programas que se insertan en la placa.

Para acabar, vamos a nombrar alguna de las posibles creaciones que podemos hacer gracias a esta pequeña placa. Podemos crear un dron volador gracias a las salidas que nos permiten controlar motores eléctricos, al igual pasa con impresoras 3D, dado que necesita motores eléctricos para desplazar el inyector de plástico. Podemos acoplar también sensores de proximidad, lo que nos permitiría hacer radares a pequeña escala. Por último vamos a poner el ejemplo de una pequeña consola, ya que aunque su hardware no sea equivalente al de los ordenadores de sobremesa, sirve para hacer tareas de mediana dificultad a la perfección.

5.2. Obtención de datos de la centralita

Hardware necesario

En este apartado vamos a explicar el hardware necesario a la hora de obtener datos del vehículo mediante el conector EOBD de este.

Primero de todo hay que decantarse por la placa que utilizaremos como base de este apartado. Por lo general en el mercado existe una gran diversidad de placas que tienen las mismas o parecidas funciones que la placa Arduino UNO, habiendo una diferencia casi nula a la hora de programar estas placas. A pesar de esto, nos decantaremos por el modelo Arduino UNO dado que ya hemos lo hemos introducido en el apartado anterior.

Podemos encontrar el Arduino UNO Rev3 (que resulta ser el más nuevo) en el distribuidor de componentes electrónicos www.electan.com por un precio de 19,45 € más 3,9 € por gastos de envío, lo que hace un total de 23,35 €.



**ARDUINO UNO ORIGINAL
CON ATMEGA328**

22.00€ **19.45€** 23.53€ con IVA

Valoración **4.8/5** (10 Votos)

REFERENCIA: AR-UNO

DISPONIBLE: **EN STOCK.** Disponible para entrega inmediata
Quedan pocas unidades!



Ilustración 59: Precio de placa modelo Arduino UNO Rev3

Llegados a este punto, no hemos mencionado en ningún momento un puerto CAN BUS, y es que este tipo de placas no vienen con estos puertos. La opción que se nos da es la de añadir “Shields” para ampliar la funcionalidad de la placa que tengamos.

Una Shield es una placa que está diseñada para aumentar las funciones de las placas con microcontroladores, en este caso el Arduino UNO. Estas funciones son muy variadas y pueden ir desde añadir conexiones tipo CAN BUS o Ethernet, hasta pantallas o GPS.

En nuestro caso buscamos una Shield que nos permita comunicar la placa Arduino con el controlador del habitáculo mediante CAN BUS. En www.amazon.com podemos encontrar numerosas Shields que implementan esta función. Nos decantaremos finalmente por el modelo MakerHawk que está por un precio de 18,99 €.



MakerHawk Escudo CAN-BUS

de MakerHawk

Sé el primero en opinar sobre este producto

Precio: EUR 19,99

Precio actual: **EUR 18,99** Envío GRATIS en pedidos superiores a 29€.

[Ver detalles](#)

Ahorras: EUR 1,00 (5%)

Precio final del producto

Compra 1 y obtén un descuento del 8% 1 promoción ▾

En stock.

Ilustración 60: Shield BUS CAN modelo MakerHawk

Una de las peculiaridades de estas Shields es que se colocan inmediatamente encima del Arduino, de forma que parece que hace de escudo, de ahí su nombre. Las Shields además emplean la totalidad de los pines de la placa Arduino, pero no por que requiera usar todos estos, si no para extender físicamente el pin, ya que con la Shield colocada encima del Arduino sería imposible acceder a estos pines.

Estos serán los únicos componentes necesarios para obtener datos de la centralita, lo que facilita enormemente su instalación.

Software empleado

A continuación vamos a ver el software necesario para poder comunicar el Arduino con el Fiat Stilo.

Primero de todo necesitamos programar el Arduino, para ello utilizaremos el programa oficial que nos facilitan desde www.arduino.cc en el apartado de software. El programa se denomina “Arduino” y su versión actual es la 1.8.5. Mediante este software tendremos total libertad a la hora de escribir en lenguaje C++ y de programar el Arduino.

Después necesitaremos una librería para poder trabajar con el Shield, ya que Arduino no está preparado nativamente para trabajar con estas y debemos de indicarle como reconocer y trabajar con la Shield que le instalemos.

Para descargarnos la librería de esta Shield debemos ir a la página web https://github.com/Seeed-Studio/CAN_BUS_Shield y a continuación hacer click en el botón “Clone or download”. Esto nos descargará un archivo .zip el cual tenemos que descomprimir en la carpeta “Library” del programa “Arduino”. Una vez realizada esta acción podremos trabajar con normalidad con la Shield. Cabe destacar que en la misma

web que hemos nombrado anteriormente se dan instrucciones claras de cómo realizar la programación inicial en caso de no tener conocimientos sobre Arduino.

Después de haber instalado en nuestro ordenador tanto Arduino como la librería para trabajar con la Shield de CAN BUS, ya podremos leer la información que circula por la red CAN del Fiat Stilo.

Conexión e instalación

Ahora que ya sabemos los componentes que necesitamos para poder leer la red CAN del Fiat Stilo, vamos a ver dónde instalarlos, como suministrarles corriente y cómo podemos conectarlos a la red del vehículo.

Para que nuestro vehículo y el arduino estén conectados, emplearemos es siguiente cable, que podemos encontrar en www.amazon.es por 7,52 €. Este nos permitirá conectar el puerto DB9 del Arduino con el puerto EOBD del Fiat Stilo, de forma que el Arduino reciba información de la red CAN del vehículo.



Sharplace Adaptador de Cable OBD2 16-Pin a DB9 9-pin Herramienta de Diagnóstico para Vehículos

de Sharplace

[Sé el primero en opinar sobre este producto](#)

Precio: ~~EUR 13,54~~

Precio actual: **EUR 7,52** Envío GRATIS.

Ahorras: EUR 6,02 (44%)

Precio final del producto

En stock.

Ilustración 61: Cable adaptador ODB2 a DB9

El siguiente paso será suministrar corriente a los componentes de nuestro sistema. Dado que la Shield se alimenta a través de los pines que dispone el Arduino, únicamente habrá que suministrar corriente al Arduino. Este se debe alimentar con 12 V en corriente continua, por lo que podemos conectarlo directamente con los pines 16 (positivo de la batería) y 4 (tierra del vehículo) del puerto EOBD del vehículo. En www.amazon.es podemos encontrar el conector Dc 2.1 por 1,74 €, que nos permitirán conexionar el puerto EOBD del vehículo con la toma de corriente del Arduino.



Adaptador de Conector Eléctrico Hembra de 5.5 x 2.1mm + Macho CCTV DC - 6 Piezas - Parte de Repuesto de Gino

★★★★★ 4 opiniones de clientes

Precio: **EUR 1,74** Envío GRATIS.

Precio final del producto

En stock.

Ilustración 62: Conector Dc 2.1

Ahora que ya tenemos todo lo necesario para leer información de la red Can del vehículo y podemos manejar dicha información con el Arduino y su software, vamos a ver dónde podemos ubicar los componentes que forman este sistema.

Dado que el Arduino junto con la Shield ocupan muy poco espacio, estos pueden ser emplazados en una de las guanteras del vehículo sin restar apenas espacio. Y el cableado que va desde el puerto EOBD hasta la guantera irá por dentro del salpicadero.

5.3. Visualización de datos a través de pantalla

En este apartado vamos a ver de qué formas podemos visualizar la información que sustraigamos de la red CAN del vehículo mediante nuestro sistema con Arduino. Para ello vamos a ver dónde podemos ubicar la pantalla, y cuál sería el software y la pantalla que mejor se adapta a nuestras necesidades.

Ubicación de pantalla

Al igual que con la primera alternativa, la pantalla que vayamos a emplear tendrá que ubicarse en la zona central del salpicadero.

Las medidas de la zona central del salpicadero son de 260 mm de largo por 165 mm de alto. La pantalla no necesita ser del tamaño máximo que disponemos en el salpicadero, ya que existen una gran cantidad de embellecedores para que el elemento que tengamos instalado en esta zona quede encuadrado y con un buen acabado.

Selección de pantalla

A continuación vamos a seleccionar la pantalla que mejor encaja en nuestras necesidades.

Buscamos una pantalla que tenga una buena resolución y que sea lo más cercana posible a las medidas antes expuestas, para poder mostrar la mayor cantidad de información posible. Obviamente debe de encontrarse a un precio razonable y también tiene que ser fácil de trabajar con ella.

Tras haber visitado las webs de varios distribuidores on-line, hemos visto que el modelo 971-GEN4-ULCD-70DT del fabricante 4D Systems es el que más tamaño tiene, que es una de las características que más nos interesa. Podemos encontrar este modelo en la web www.mouser.es por un precio de 126,65 €.

gen4-uLCD-70DT		En existencias: 77	
 <p>Agrandar</p> <p>Las imágenes son sólo para</p>	Mouser REF:	971-GEN4-ULCD-70DT	
	Fabr. N.º:	gen4-uLCD-70DT	
	Fabr.:	4D Systems	
	Ref. Cliente:	<input type="text"/>	
	Descripción:	Módulos de pantalla 7.0". 800x480 pixels Slim Display Module	
Ciclo de vida:	 Nuevo producto: Novedad de este		
		Existencias:	77 Puede enviarse inmediatamente
		Pedido:	0
		Plazo de producción de fábrica:	6 Semanas
		Introducir cantidad:	Mínimo: 1 Múltiples: 1 <input type="text" value="1"/> <input type="button" value="Comprar"/>
		Precio unitario:	126,65 €

Ilustración 63: Pantalla modelo 971-GEN4-ULCD-70DT

Este modelo de pantalla cuenta con 7 pulgadas de tamaño, una resolución de 800 por 480 píxeles y con un táctil tipo resistivo. La resolución de esta pantalla no es muy elevada, pero ha sido imposible encontrar otra pantalla que tengan mayores características en este aspecto que la seleccionada.^{14 15}

Instalación de pantalla

A la hora de conectar la pantalla al Arduino, necesitaremos una Shield, ya que la pantalla no se puede conectar al Arduino nativamente. La Shield que necesitamos la podemos encontrar en la misma web y fabricante que la pantalla por un precio de 10,20 €.


4D Arduino Adaptor Shield		En existencias: 6	
	Mouser REF:	971-ARDUINOADPTRSHLD	
	Fabr. N.º:	4D Arduino Adaptor Shield	
	Fabr.:	4D Systems	
	Ref. Cliente:	<input type="text"/>	
	Descripción:	Enchufes y adaptadores Adaptor Board for Arduino	
		Existencias:	6 Puede enviarse inmediatamente
		Pedido:	0
		Mínimo:	1 Múltiples: 1
		Introducir cantidad:	<input type="text" value="1"/> <input type="button" value="Comprar"/>
		Precio unitario:	10,20 €
		Precio total:	10,20 €

Ilustración 64: Shield adaptador para pantalla

A pesar de ya tener una Shield instalada en el Arduino, podemos acoplar otra encima sin mayor problema dado que como ya sabemos, la primera Shield puentea todos los pines del Arduino, haciéndolos accesibles para otros añadidos.¹³

La única conexión que requiere la pantalla es un cable compuesto de 5 vías que viene incluido con la pantalla. Con este conectaremos la pantalla a la Shield y podremos suministrar los datos y la corriente necesarios para que la pantalla funcione.

Además de estos componentes, necesitaremos también una tarjeta microSD, para poder almacenar los programas que elaboremos en la pantalla. Podemos encontrar en www.amazon.es una tarjeta microSD de 16 GB por 8,19 €.



Kingston SDC10G2/16GB - Tarjeta de memoria microSD, 16 GB, clase 10 UHS-I, 45 MB/s, color negro
de Kingston

★★★★☆ 884 opiniones de clientes | 89 preguntas respondidas

Amazon's Choice de "micro sd 16 gb"

Precio recomendado: EUR 24,95
Precio: **EUR 8,19** Envío GRATIS en pedidos superiores a 29€. [Ver detalles](#)

Ahorras: **EUR 16,76 (67%)**
Precio final del producto

Ilustración 65: Tarjeta microSD

Hay que recalcar que las Shields se alimentan de corriente directamente del Arduino, por lo que no es necesario alimentarlas externamente.

Con todos estos elementos, podemos conectar sin mayor problema la pantalla al Arduino.

Software empleado

Llegados a este punto, vamos a ver qué tipo de software necesitamos para poder programar esta pantalla.

Necesitaremos para empezar instalar una librería con tal de que el Arduino reconozca la nueva Shield y la pantalla que hemos instalado. Podemos encontrar todas las librerías que necesitamos en la siguiente dirección web:

http://www.4dsystems.com.au/product/4D_Workshop_4_IDE/downloads

Ahora necesitamos darle imagen al contenido que queremos mostrar por pantalla. Para ello emplearemos el programa denominado "Workshop4" que podemos encontrar gratis en la misma dirección web. Con este podremos tener total libertad a la hora de mostrar contenido a través de la pantalla.¹⁶

Con esto ya tenemos todo listo para programar con nuestro sistema.

5.4. Costes y conclusión

Como ya hemos visto con la primera alternativa, los últimos pasos son ver el coste total de la alternativa y analizar esta desde varios aspectos.

Por lo tanto primero expondremos una tabla donde mostraremos los elementos que hemos empleado para la elaboración de nuestro sistema, así como su coste. Cabe destacar que los precios de todos los productos pueden variar en función del tiempo. Los precios de los componentes que mostramos a continuación datan de Diciembre de 2017.

Componente	Distribuidor	Precio (€)
Arduino UNO Rev3	Electan	23,35
Shield MakerHawk CAN-BUS	Amazon	18,99
Cable adaptador OBD2 a DB9	Amazon	7,52
Conector Dc 2.1	Amazon	1,74
Pantalla 971-GEN4-ULCD-70DT	Mouser	126,65
Shield adaptador para pantalla	Mouser	10,2
Tarjeta microSD 16 GB	Amazon	8,19
Precio Total (€)		196,64

Tabla 6: Costes de la segunda alternativa: Arduino

Ahora que ya tenemos desglosado el coste de todos los materiales, vamos a valorar y a sacar conclusiones de los distintos factores relevantes que afectan a la elaboración de todo este sistema. Iremos desde su instalación en el Fiat Stilo, pasando por su programación, hasta llegar a la muestra de información por pantalla.

-Instalación en vehículo: La instalación en el vehículo es muy simple dado que la única conexión que requiere el Arduino es al puerto EOBD del controlador del habitáculo, que le suministra tanto información como corriente eléctrica. En este caso no se necesitan ni convertidores para la transmisión de datos ni transformadores para la corriente, por lo que todo el sistema funciona de una forma más directa.

-Seguridad: Como ya mencionamos anteriormente, la seguridad es un apartado muy delicado, ya que trabajamos con información sensible. Aquí el Arduino no cuenta con medidas de seguridad que le protejan de condiciones extremas, tampoco cuenta con un software a prueba de fallos que evite el cuelgue del sistema o el manejo de información errónea.

-Configuración inicial: Antes de comenzar a programar hay que realizar una serie de configuraciones con tal de que los componentes de nuestro sistema puedan trabajar a la perfección. Este paso es bastante sencillo dado que solamente requiere el uso de las respectivas librerías de cada Shield y de unas pocas líneas de programación para que el Arduino sea capaz de trabajar con las Shields y la pantalla.

-Programación: Una vez está todo configurado e instalado, hay que tener conocimientos sobre programación para extraer la información que deseamos del Fiat Stilo. Pero hoy

en día es muy fácil aprender sobre programación dado la alta popularidad y extensión de Arduino y derivados de este.

-Visualización por pantalla: El software “4D Workshop 4 IDE” que empleamos para diseñar el programa de la pantalla nos deja a nuestra disposición una gran cantidad de instrumentos con los que poder trabajar. De este modo tenemos mucha libertad a la hora de cómo mostrar información por pantalla.

-Coste en tiempo: La instalación en el vehículo es bastante sencilla y aunque su programación inicial puede ser algo engorrosa, lo compensa con mucha facilidad a la hora de diseñar la interfaz gráfica que se muestra por pantalla.

-Coste: En este apartado cabe destacar lo barato que resulta hacer este tipo de proyectos mediante Arduino. Esto se debe mayormente a que tanto el hardware como el software que empleamos son libres y de mucha variedad, lo que ayuda a reducir el precio de cada componente empleado en nuestro sistema.

Ahora que ya tenemos las conclusiones de todos los apartados relevantes, vamos a valorarlos de 1 al 10 de cara a futuras comparativas con los sistemas que funcionan mediante autómatas y Raspberry Pi. Para esta comparativa nos basaremos en la siguiente tabla, donde al final compararemos las 3 alternativas.

	Instalación en vehículo	Seguridad	Configuración inicial	Programación	Visualización por pantalla	Coste en tiempo	Coste	Valoración Final
Valoración	7	3	8	5	9	4	8	44

Tabla 7: Valoración de la alternativa Arduino

6. Solución alternativa: Raspberry

6.1. Introducción a la Raspberry

Nuestra tercera y última alternativa se trata de la Raspberry Pi, que guarda cierta semejanza con el Arduino. A continuación explicaremos brevemente su historia y posteriormente expondremos sus características así como algunos ejemplos de las posibilidades que nos ofrece.

La Raspberry viene a ser un ordenador de placa reducida, donde todos sus componentes están ensamblados a una placa de circuitos impresos más pequeña que una tarjeta de crédito. Esta placa no es de hardware libre aunque permite la modificación de sus componentes. Su software sí que es libre y permite instalar una gran variedad de sistemas operativos, estando entre ellos variantes de Windows y Linux.

El concepto de la Raspberry fue creado en 2006 por David Braven, un antiguo diseñador de videojuegos. Su objetivo con esta creación era que los niños aprendieran el funcionamiento básico de los ordenadores y que se divirtieran con ello.

Posteriormente Eben Upton se unió al proyecto y comenzó a darle forma desarrollando tanto el software como el hardware de la Raspberry.

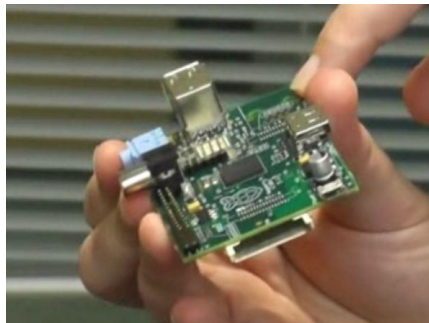


Ilustración 66: Prototipo de la primera Raspberry

En 2011 se comenzaron a producir a cuentagotas los modelos alfa y posteriormente los beta. Una vez solventado algún que otro fallo de estas versiones, se puso en marcha la producción en masa de estas versiones en China, abaratando así costes y tiempos de producción. Finalmente se pusieron en venta estas unidades a principios de 2012, agotándose en pocos minutos las unidades existentes.

Como hemos visto, existen varios modelos de Raspberry Pi. Nos centraremos en explicar los componentes de la más reciente, la Raspberry Pi 3 Model B.

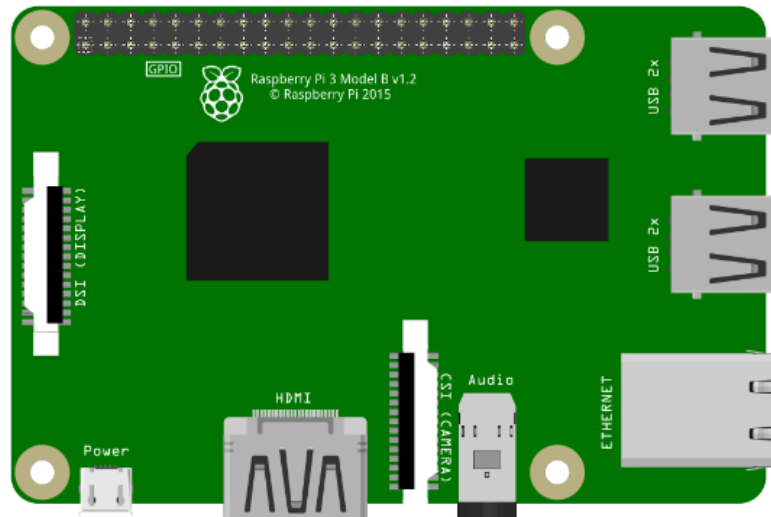


Ilustración 67: Esquema de la Raspberry Pi 3 Model B

-CPU: Es el cerebro de toda la placa, el que se encarga de procesar toda la información que la placa envía o recibe. En este modelo el procesador tiene 4 núcleos y funciona a 1.2 GHz.

-GPU: Este chip es el que se encarga de gestionar toda la información que se muestra por la pantalla que tengamos conectada, siendo capaz de mover contenido multimedia de calidad Blu-ray.

-Alimentación: La Raspberry se alimenta a través de un puerto micro USB ubicado en la parte inferior izquierda de la placa. Esta consume 5 V a corriente continua y 3 A.

-HDMI: La Raspberry Pi se puede conectar directamente a un monitor y es capaz de sacar una señal de video de hasta 1920 por 1200 pixeles a 30 fotogramas por segundo. Este es uno de sus puntos fuertes dado que no requiere de elementos externos para sacar imagen y además el puerto empleado está estandarizado y se encuentra en la totalidad de televisores y monitores del mercado.

-Jack de 3,5 mm: Otra característica muy relevante de la Raspberry es que tiene una conexión directa para el audio, que nos permite reproducir sonidos a través de auriculares o altavoces. Hay que destacar que el conector HDMI también puede ser configurado para que además de transferir imagen, transfiera también sonido.

-Puerto Ethernet: Esta placa tiene la opción de conectarse a internet o a una red mediante una conexión Ethernet. No es necesario por tanto ningún tipo de Shield o adaptador.

-Puerto USB: Este modelo tiene un total de 4 puertos USB donde podemos conectar sin ningún problema ratones, teclados, memorias usb, y todo tipo de periféricos.

-GPIO: Al igual que Arduino, Raspberry también tiene entradas y salidas de este tipo, que le proporcionan una infinidad de posibilidades a la hora de conectar todo tipo de sensores.

-Lector de tarjetas: Dado que la Raspberry no tienen almacenamiento interno, se hace necesario un lector de tarjetas para leer tarjetas MicroSD donde irán almacenados tanto el sistema operativo como las distintas aplicaciones.

-RAM: Esta es la memoria que tiene la Raspberry para almacenar los datos de las aplicaciones que están activas. Este modelo en concreto cuenta con 1 GB.

A simple vista la Raspberry cuenta con componentes que podemos ver en cualquier ordenador hoy en día, y es que realmente es un microordenador. Debido a esto se pueden realizar un sinnúmero de proyectos, pudiendo hacer desde un centro multimedia o una consola hasta un escáner en 3D gracias a la posibilidad de incluir una cámara en la Raspberry.^{17 18}

6.2. Obtención de datos de la centralita

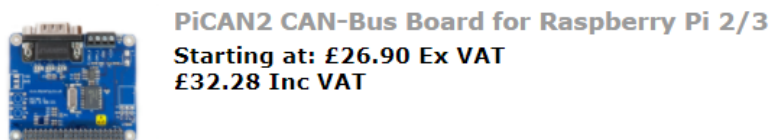
Hardware necesario

Ya hemos aclarado que el modelo de Raspberry empleado será el Raspberry Pi 3 Model B por ser el más moderno y completo. Podemos encontrar este modelo muy económico en www.amazon.com por 33,74 €.



Ilustración 68: Raspberry Pi 3 Model B

Ahora que ya tenemos el modelo de Raspberry, vamos a ver cómo podemos conectar esta a la red Can del vehículo. Existe una Shield para Raspberry cuya función es la de leer y escribir información en una red CAN. Esta Shield se denomina PiCAN 2 y la podemos encontrar en www.skpang.go.uk por 32,28 £, o lo que es lo mismo, 36,46 €. ¹⁹

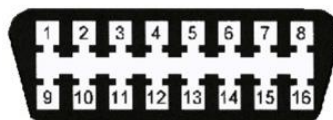


Con esta Shield acoplada a la Raspberry, ya podremos recibir información de la red CAN del Fiat Stilo a través del puerto EOBD del controlador del habitáculo. En la siguiente imagen podemos ver cómo quedan la Shield y la Raspberry ensambladas.



Ilustración 69: Raspberry junto con la Shield PiCAN 2

A continuación vamos a explicar cómo se conectaría la red CAN del vehículo con la Shield. Para ello antes hay que comentar que la Shield necesita de alimentación externa. Dicho esto podemos ver en la imagen anterior que la Shield tiene un conector DB9 para conectar esta directamente con el conector OEDB del Fiat Stilo, pero a su lado tiene también 4 terminales atornillables. Estos terminales corresponden a los canales CAN_H y CAN_L de la red CAN del vehículo, a la tierra del vehículo, y a la alimentación de 12 V en corriente continua que requiere la Shield para trabajar. Optaremos por estos terminales dado que en posteriormente necesitamos suministrar corriente a la Raspberry también desde el puerto EOBD, por lo tanto un cable multifilar del puerto EOBD al DB9 de la Shield nos impediría esta labor.



1 - Sin uso	9 - Sin uso
2 - J1850 Bus positivo	10 - J1850 Bus negativo
3 - Sin uso	11 - Sin uso
4 - Tierra del Vehículo	12 - Sin uso
5 - Tierra de la Señal	13 - Tierra de la señal
6 - CAN High	14 - CAN Low
7 - ISO 9141-2 - Línea K	15 - ISO 9141-2 - Línea L
8 - Sin uso	16 - Batería - positivo

Ilustración 70: Pines del conector EOBD

Recalamos nuevamente esta imagen para ver los pines que tendríamos que emplear. Por tanto las conexiones se deberán de hacer del siguiente modo: Pin 6 con terminal CAN_H, pin 14 con terminal CAN_L, pin 4 con la tierra y pin 16 con el polo positivo de la batería, que recordemos que funciona a 12 V en corriente continua.

Una vez hemos alimentado de corriente y hemos conectado a la red la Shield, pasamos a alimentar la Raspberry Pi. En la descripción de la Raspberry hemos visto que requiere 5 V en corriente continua y 3 A para su correcto funcionamiento, y dado que el pin 16 de la batería nos suministra 12 V, necesitaremos un transformador que convierta esos 12 V en 5 V.

Hemos escogido el siguiente transformador de www.amazon.es ya que nos suministra 5 V en corriente continua a 3 A, que será suficiente para la Raspberry. Todo a un precio razonable de 7,01 €.



JZK® Coche Poder Convertidor DC 12V a 5V / 3A voltaje Convertidor Con doble USB Adaptador Conectores Para la carga del teléfono coche audio radio Etc de JZK
[Sé el primero en opinar sobre este producto](#)

Precio: **EUR 7,01** Envío **GRATIS** en pedidos superiores a 29€. [Ver detalles](#)
 Precio final del producto

Ilustración 71: Transformador de 12 V a 5 V

Este transformador irá conectado a los pines 4-Tierra del vehículo y 16-Positivo de la batería del puerto EOBD

Mediante un conector o macho de EOBD conseguiremos tener acceso a los pines 6-CAN High, 14-CAN Low, 4-Tierra del vehículo y 16-Positivo de la batería. Podemos conseguir un conector de este tipo muy fácilmente a través de internet. Sin ir más lejos tenemos este de www.amazon.es por 2,83 €:



MagiDeal OBD 2 Conector 16 Pines Adaptador Conectores Macho Herramienta para Diagnóstico de Magideal
[Sé el primero en opinar sobre este producto](#)

Precio: ~~EUR 5,09~~
 Precio actual: **EUR 2,83** Envío **GRATIS**.
 Ahorras: **EUR 2,26 (44%)**
 Precio final del producto

Ilustración 72: Conector macho puerto EOBD

Ahora conectaremos el transformador a la Raspberry mediante un cable micro USB. Podemos encontrar en www.amazon.es cables de este tipo por 8,49 €.



3m Premium Cable MicroUSB a USB de alta velocidad | Nylon trenzado | Cable cargador y de datos | Cable de carga rápida | Contactos recubiertos de oro 24 k | para Android, Samsung, HTC, Motorola, Nokia, LG, HP, Sony, Blackberry y más | plateado de UpLink
 ★★★★★ 187 opiniones de clientes | 17 preguntas respondidas
Amazon's Choice de "cable micro usb 3m"

Precio: **EUR 8,49** Envío **GRATIS** en pedidos superiores a 29€. [Ver detalles](#)
 Precio final del producto

Ilustración 73: Cable micro USB

Finalmente queda instalar una tarjeta microSD en la Raspberry para tener suficiente espacio donde almacenar y ejecutar los programas que queramos, así como el sistema operativo de la propia Raspberry. Podemos encontrar en www.amazon.es una tarjeta microSD de 16 GB por 8,19 €.



Kingston SDC10G2/16GB - Tarjeta de memoria microSD,
16 GB, clase 10 UHS-I, 45 MB/s, color negro

de Kingston



884 opiniones de clientes | 89 preguntas respondidas

Amazon's Choice de "micro sd 16 gb"

Precio recomendado: EUR 24,95

Precio: **EUR 8,19** Envío GRATIS en pedidos superiores a 29€. Ver detalles

Ahorras: EUR 16,76 (67%)

Precio final del producto

Ilustración 74: Tarjeta MicroSD

Con todo esto ya tendremos el sistema preparado para comenzar a programar y recibir información de la red CAN del Fiat Stilo.

Software necesario

Ahora vamos a ver el software que necesitamos para poner en funcionamiento todo el sistema que tenemos y así poder manejar la información que recibamos de la red CAN del Fiat Stilo.

Primero de todo necesitamos tener instalado en la tarjeta MicroSD de la Raspberry el sistema operativo. Hemos elegido como sistema operativo el Raspbian.

Tendremos que ir a la sección “downloads” de la página web oficial de Raspberry www.raspberrypi.org, hacer click en “NOOBS” y descargar la versión .zip del “NOOBS”. Esta descarga contiene ya el sistema operativo de la Raspberry. Lo que tenemos que hacer para instalarla en esta, es copiar los datos que nos hemos descargado directamente en la tarjeta MicroSD e insertar la tarjeta en la Raspberry.

Encenderemos la Raspberry y con la ayuda de una pantalla, un teclado y un ratón seguiremos los sencillos pasos que se requieren para que la instalación del sistema operativo se lleva a cabo.

Con la Raspberry Pi encendida, iremos a las opciones y cambiaremos el inicio a “a CLI”. Esto hará que cuando reiniciemos la Raspberry, esta entre en modo CLI, de forma que podemos modificar una serie de cosas para que la Raspberry pueda entender la red CAN del vehículo.

Una vez reiniciada la Raspberry, escribiremos lo siguiente:

```
Sudo nano /boot/config.txt
```

Esto abrirá un archivo tipo .txt. En este archivo añadiremos al final las 3 siguientes líneas:

```
dtoverlay=spi=on
```

```
dtoverlay=mcp2515-can0,oscillator=16000000,interrupt=25
```

```
dtoverlay=spi-bcm2835-overlay
```

Ahora reiniciaremos la Raspberry mediante el comando “sudo reboot”. Y a continuación introduciremos el comando “sudo /sbin/ip link set can0 up type can bitrate 500000”.²⁰

Para finalizar esta parte instalaremos en la Raspberry el contenido de la página web <https://github.com/linux-can/can-utils>.

Ahora seguiremos unos últimos pasos para instalarnos en la Raspberry el programa Python-CAN y así poder finalmente recibir información de la red CAN del vehículo.

Nos descargaremos el contenido de la siguiente dirección web y lo descomprimiremos directamente en la tarjeta MicroSD:

<https://bitbucket.org/hardbyte/python-can/get/4085cffd2519.zip>

Una vez iniciada la Raspberry en el modo CLI, volveremos a introducir el código “sudo /sbin/ip link set can0 up type can bitrate 500000” e instalaremos el contenido que acabamos de descargar escribiendo el comando “sudo python3 setup.py install”. Ejecutaremos el programa escribiendo el comando “python3” y escribiremos las siguientes líneas:

```
import can

bus = can.interface.Bus(channel='can0', bustype='socketcan_native')

notifier = can.Notifier(bus, [can.Printer()])
```

Una vez realizados todos estos pasos, nos aparecerán por pantalla los mensajes que circulan por la red CAN del Fiat Stilo. Seremos por tanto capaces de recibir información desde la red CAN del vehículo.^{22 23}

Instalación de los componentes

Como ya hemos visto en los mismos apartados de las otras dos alternativas, existe espacio de sobra en las guanteras para instalar y almacenar todos los componentes. De esta forma siempre podemos tener acceso cuando queramos a toda la instalación y además quedarán ocultos a simple vista.

El cableado requerido para este sistema está explicado en el apartado “hardware necesario”, así que no será necesario volver a nombrarlo. Hay que remarcar que todo el cableado irá por dentro del salpicadero, desde el controlador del habitáculo, donde se encuentra el puerto EOBD, hasta la zona de las guanteras, donde se encuentran tanto la Shield como la Raspberry.

6.3. Visualización de datos a través de pantalla

En este apartado vamos a ver de qué formas podemos visualizar la información que sustraigamos de la red CAN del vehículo mediante nuestro sistema con Raspberry. Para ello vamos a ver dónde podemos ubicar la pantalla, y cuál sería el software y la pantalla que mejor se adapta a nuestras necesidades.

Ubicación de pantalla

Al igual que con la primera y segunda alternativa, la pantalla que vayamos a emplear tendrá que ubicarse en la zona central del salpicadero.

Las medidas de la zona central del salpicadero son de 260 mm de largo por 165 mm de alto. La pantalla no necesita ser del tamaño máximo que disponemos en el salpicadero,

ya que existen una gran cantidad de embellecedores para que el elemento que tengamos instalado en esta zona quede encuadrado y con un buen acabado.

Selección de pantalla

Ahora buscaremos una pantalla idónea que se ajuste a nuestras características. Buscamos entre otras cosas que sea del mayor tamaño posible y con una buena resolución, a la vez que también necesitamos que sea táctil, para así evitar el uso de ratón y teclado para el uso cotidiano de esta pantalla. Esta búsqueda se ve facilitada en gran medida debido a que la Raspberry se puede conectar a una pantalla a través de un puerto HDMI, que es un estándar actualmente, por lo que todas las pantallas suelen tener uno.

En www.amazon.es podemos encontrar una pantalla de la marca Makibes de 7 pulgadas de tamaño, 1024 por 600 pixeles por pulgada, con función táctil y puerto HDMI. Esta tiene un precio de 59,99 €.



Ilustración 75: Pantalla Makibes 7 pulgadas

Instalación de pantalla

El siguiente paso es conectar la pantalla a la Raspberry para la transmisión de imagen y a su vez, suministrarle corriente eléctrica para su funcionamiento.

Una de las ventajas de haber elegido esta pantalla es que se alimenta a través de un conector microUSB y que por lo tanto requiere 5 V en corriente continua. Dado que nuestro modelo de Raspberry tiene 4 puertos USB, esta puede suministrarle corriente a la pantalla sin problemas.

Respecto a la transmisión de imagen, necesitaremos un simple cable HDMI para conectar los puertos HDMI de la pantalla y la Raspberry y así conseguir señal de video.

Dicho esto solamente necesitaremos un cable HDMI y un cable de microUSB a USB para que la pantalla sea plenamente funcional. Estos cables vienen incluidos en la caja de la pantalla, por lo que no será necesario adquirirlos.

Software empleado

Por último queda ver cómo vamos a mostrar la información por pantalla. Tenemos que encontrar un software o aplicación que nos permita diseñar una interfaz que muestre los datos que obtenemos de la red CAN de una forma fácil y atractiva.

Para esta tarea emplearemos una herramienta de programación que viene incluida ya en el sistema operativo que hemos instalado previamente en la Raspberry. Se trata de Python, un programa mediante el cual podremos crear una interfaz gráfica de usuario o GUI (Graphical User Interface). Gracias a esta GUI interactuaremos de forma intuitiva

con la Raspberry y podremos llegar a controlar y visualizar los parámetros que deseemos del Fiat Stilo.

Como ya hemos dicho antes, Python viene incluido con el sistema operativo de la Raspberry, por lo que no hará falta descargar ningún software ni librería para comenzar a diseñar la GUI.^{21 24 26}

6.4. Costes y conclusión

Como ya hemos visto con la primera y segunda alternativa, los últimos pasos son ver el coste total de la alternativa y analizar esta desde varios aspectos.

Por lo tanto primero expondremos una tabla donde mostraremos los elementos que hemos empleado para la elaboración de nuestro sistema, así como su coste. Cabe destacar que los precios de todos los productos pueden variar en función del tiempo. Los precios de los componentes que mostramos a continuación datan de Diciembre de 2017.

Componente	Distribuidor	Precio (€)
Raspberry Pi 3 Model B	Amazon	33,74
Shield PiCAN2	skpang.go.uk	36,46
Transformador 12 V a 5 V	Amazon	7,01
Conector EOBD	Amazon	2,83
Cable micro USB	Amazon	8,49
Tarjeta microSD 16 GB	Amazon	8,19
Pantalla Makibes 7"	Amazon	59,99
Precio Total (€)		156,71

Tabla 8: Coste de la tercera Alternativa: Raspberry

Una vez hemos determinado el coste total de la alternativa, vamos a valorar y a sacar conclusiones de los distintos factores relevantes que afectan a la elaboración de todo este sistema. Iremos desde su instalación en el Fiat Stilo, pasando por su programación, hasta llegar a la muestra de información por pantalla.

-Instalación en vehículo: La instalación de la Raspberry en el Fiat Stilo es sencilla dado que solamente necesitaremos emplear un número reducido de pines del puerto EOBD, que irán conectados o al transformador para suministrar corriente a la Raspberry o a Shield para que nuestro sistema se pueda comunicar con la red CAN del Fiat Stilo.

-Seguridad: Como ya mencionamos anteriormente, la seguridad es un apartado muy delicado, ya que trabajamos con información sensible. En este caso la Raspberry no cuenta con medidas de seguridad que le protejan de condiciones extremas, a la vez que tampoco cuenta con un software a prueba de fallos que evite el cuelgue del sistema o el manejo de información errónea.

-Configuración inicial: Antes de comenzar a programar se requiere tanto la instalación del sistema operativo de la Raspberry como realizar una serie de configuraciones

iniciales con el objetivo de que nuestro sistema trabaje correctamente y se pueda comunicar con la red CAN de Fiat Stilo. A pesar de se requieren acciones sencillas, esto puede llevar tiempo a gente que no esté familiarizada con la programación y los sistemas operativos.

-Programación: Con nuestro sistema configurado y listo para funcionar, se debe de tener conocimientos sobre programación para comunicarnos con la del Fiat Stilo y para manejar la información que recibamos. A pesar de necesitar conocimientos sobre programación, existen una gran cantidad de proyectos por todo internet de los cuales podemos tomar ejemplo a la hora de elaborar nuestro programa.

-Visualización por pantalla: Como hemos visto antes, mediante el Python que ya tenemos instalado de serie en el sistema operativo podremos comenzar a crear la GUI. Por el contrario en esta alternativa no tenemos a nuestro alcance tantos instrumentos como nos aportaban el Tia Portal o el 4D Workshop 4 IDE de las anteriores alternativas.

-Coste en tiempo: La instalación en el vehículo es sencilla aunque la creación de la GUI por otro lado será algo lenta dado que la tenemos que crear desde 0.

-Coste: El coste total de esta alternativa es realmente bueno debido a que no solo la Raspberry y la Shield son baratas, si no que la pantalla que es de buena calidad y no encarece el precio final de la alternativa. Este abaratamiento se ha podido conseguir dado que la Raspberry emplea conexiones estándar para la transmisión de video, lo que aumenta muchísimo el rango de pantallas que se ajustan a las necesidades de esta alternativa.

Ahora que ya tenemos las conclusiones de todos los apartados relevantes, vamos a valorarlos de 1 al 10 de cara a futuras comparativas con los sistemas que funcionan mediante autómatas y Arduino. Para esta comparativa nos basaremos en la siguiente tabla, mediante la cual al final compararemos las 3 alternativas.

	Instalación en vehículo	Seguridad	Configuración inicial	Programación	Visualización por pantalla	Coste en tiempo	Coste	Valoración Final
Valoración	6	3	7	5	8	3	9	41

Tabla 9: Valoración de la alterativa Raspberry

7. Solución adoptada

Llegados a este punto y habiendo estudiado ya las tres alternativas, vamos a compararlas y a escoger una de estas.

A lo largo de este TFG hemos elaborado una tabla para cada alternativa donde se valoran de 1 a 10 cada una de las características más destacables. Esta tabla tiene como objetivo final el ayudarnos a comparar las tres alternativas de forma sencilla. A continuación mostramos las tres tablas unidas en una.

Alternativa	Instalación en vehículo	Seguridad	Configuración inicial	Programación	Visualización por pantalla	Coste en tiempo	Coste	Valoración Final
Autómata	3	10	7	7	9	5	2	43
Arduino	7	3	8	5	9	4	8	44
Raspberry	6	3	7	5	8	3	9	41

Tabla 10: Comparativa de las tres alternativas estudiadas

Lo primero que llama a la vista de esta tabla es la valoración final de las alternativas, donde hay una diferencia de 3 puntos entre la peor y la mejor alternativa. Cabe destacar que la máxima puntuación posible es de 70 puntos, por lo que la diferencia que hemos mencionado anteriormente es mínima. También es remarcable que todas las alternativas han sobrepasado los 35 puntos, lo que sería un 5 en una escala de 0 a 10, siendo estas puntuaciones aceptables.

Otra cosa que vamos a ver mediante esta tabla son las características diferenciadoras de cada característica, ya sean para bien o para mal. Explicaremos el porqué de esta diferencia basándonos en los conocimientos adquiridos a lo largo de este TFG.

Podemos ver como en el apartado “Instalación el vehículo” el autómata ha obtenido menor puntuación. Esto se debe a que hay que alimentar mediante corriente eléctrica a un total de 4 componentes mediante transformadores, además de establecer una red mediante un switch para poder establecer una red de comunicación. Las otras dos alternativas tienen mayor puntuación ya que se requiere suministrar corriente a un menor número de elementos, así como tampoco existen tantos componentes que requieren conectarse entre sí.

Otro aspecto muy destacable e importante es el la “Seguridad”. Aquí la opción del autómata obtiene la máxima puntuación dado que el autómata, como ya dijimos anteriormente, está preparado para trabajar en las peores condiciones, propias de una industria. Además cuenta con un software muy simple creado así con el objetivo de que el sistema sea más estable y se eviten así posibles cuelgues, que en una industria supondrían grandes pérdidas. Las otras dos alternativas, a pesar de que tengan un hardware con una potencia de procesado considerable, ni garantizan su trabajo en condiciones críticas, ni nos protegen frente a cualquier posible cuelgue o interferencia.

Otro apartado a destacar es el de “Programación”. En este el autómata vuelve a posicionarse por delante de las dos otras alternativas, debido principalmente a que trabajamos con un software cerrado y preparado como el Tia Portal, con el que trabajaremos el autómata de la marca Siemens. Gracias a este programa podemos diseñar tanto la interfaz gráfica como los procesos que debe ejecutar el autómata.

El último apartado a mencionar sería el de “Coste”. Como vemos la alternativa del autómeta tiene un elevado coste y se refleja en esta puntuación. Esta alternativa tiene un precio de 693,47 € frente a los 196,64 € de la alternativa del Arduino o los 156,71 € de la Raspberry. El elevado coste de la alternativa del autómeta se debe principalmente a que necesitamos más elementos para el funcionamiento del sistema, y que estos a su vez son más caros ya que están muy bien protegidos frente agentes y perturbaciones externos.

Una vez vistas las valoraciones con puntuaciones más dispares, vamos a proceder a nombrar la mejor alternativa, así como el porqué de esta elección.

La Raspberry es la alternativa que peor puntuación ha obtenido con un total de 41 puntos, que se aproxima mucho a las otras dos alternativas. Por otro lado el Arduino ha sido el que mayor puntuación ha obtenido con un total de 44 puntos, que está ligeramente por encima de los 43 puntos que ha obtenido el autómeta.

Visto esto, la elección obvia sería el Arduino, aunque como ya dijimos al principio de este TFG, nos decantaremos finalmente por la opción que más seguridad nos aporte, siendo esta el autómeta. Esta elección se debe a que aparte de no haber muchas diferencias respecto a la puntuación final (refiriéndonos al autómeta respecto del Arduino), la seguridad es muy importante en un proyecto que trabaja con información tan delicada como puede ser la de un vehículo en circulación. Ya que en caso de haber algún fallo o interferencia a la hora de transmitir/recibir información en la red del vehículo, supondría un riesgo para los ocupantes del vehículo y también para las personas cercanas a dicho vehículo. Es por este motivo que seleccionamos el autómeta industrial como nuestra mejor alternativa, ya que las otras dos alternativas no aportan nada de seguridad en comparación con el autómeta.

8. Implementación de la aplicación

8.1. Introducción

Una vez hemos elegido el autómatas como base para nuestro proyecto, vamos a mostrar mediante un ejemplo todas las posibilidades que el autómatas nos ofrece.

Para poder poner en marcha este ejemplo, ha sido necesario trabajar con un autómatas cedido por el Departamento de Ingeniería Eléctrica de la Universidad Politécnica de Valencia. Por lo tanto este ejemplo se ha realizado en una de las aulas de dicho departamento.

Tendremos a nuestra disposición el autómatas Siemens modelo S7 1214C AC/DC/RLY que es bastante similar al S7 1212C DC/DC/RLY que hemos escogido nosotros para nuestra alternativa. Una diferencia entre estos es que el modelo 1212C es alimentado a través de corriente continua y en 1214C por corriente alterna, característica que solo influye a la hora de la instalación del autómatas. Otra diferencia son las entradas/salidas de cada uno, teniendo el 1212C 8 entradas y 6 salidas digitales frente a las 14 entradas y 10 salidas digitales del 1214C. La última diferencia remarcable es que la memoria de trabajo del modelo 1212C es de 75 kb frente a los 100 kb de memoria de trabajo del 1214C. Aunque existen diferencias entre ambos, ninguna supone problema alguno a la hora de manejar toda la información que se requiere.

Dicho esto, vamos a ver alguna de las modificaciones que se le han realizado al autómatas de nuestro ejemplo con el objetivo de que sea más intuitivo y didáctico trabajar con él.



Entradas
analógicas

Entradas
digitales

Salidas

Ilustración 76: Autómatas del laboratorio

Como indicamos en la foto, en la parte superior del autómatas tenemos las entradas analógicas y digitales. De esta forma tenemos 2 entradas analógicas que se manejan mediante potenciómetros, 2 entradas digitales que se accionan mediante pulsadores, y el resto que se accionan mediante palancas.

Inmediatamente debajo del autómatas tenemos una serie de leds. Cada uno de estos corresponde a cada una de las salidas que tiene el autómatas.

Cómo podemos ver después de esta descripción, el autómatas viene ya preparado para que nosotros podamos acceder tanto a las entradas como a las salidas de este sin mayor problema, lo que es ideal para trabajar este ejemplo.

El otro elemento que tenemos a parte del autómatas es el nuestro portátil, que hará como pantalla o HMI. Las únicas características relevantes son que dicho portátil emplea una resolución de 1366 por 768 píxeles y que su pantalla no es táctil, por lo que emplearemos un ratón para esta función.

Llegados a este punto ya hemos descrito los dos únicos elementos que necesitamos para el correcto funcionamiento de este ejemplo.

Hay que recalcar que esto es un mero ejemplo de lo que se puede hacer con nuestro sistema. Las posibilidades son casi infinitas dado que en la realidad tendríamos acceso a todas las variables que circulan por la red CAN del vehículo. Por tanto en este ejemplo vamos a mostrar algunas de las funciones más típicas con las que podemos trabajar.

Ahora vamos a explicar cómo hemos hecho la configuración inicial, los programas o procesos que hemos creado, y su funcionamiento final.

8.2. Configuración inicial

Una vez iniciado el Tia Portal V14 en nuestro portátil y creado un proyecto, lo primero que debemos hacer es añadir nuestro autómatas. Para ello haremos click en "Agregar dispositivo" y en la ventana emergente iremos a Controladores → SIMATIC S7-1200 → CPU → CPU 1214C AC/DC/RLY y aquí seleccionaremos el modelo 6ES7 214-1BG40-0XB0 y haremos click en aceptar.

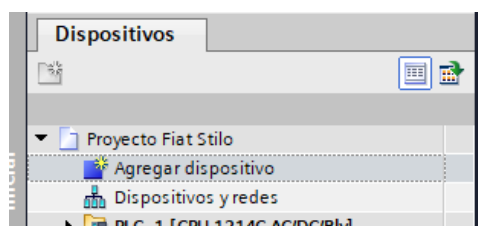


Ilustración 77: Opción "Agregar dispositivo" en Tia Portal

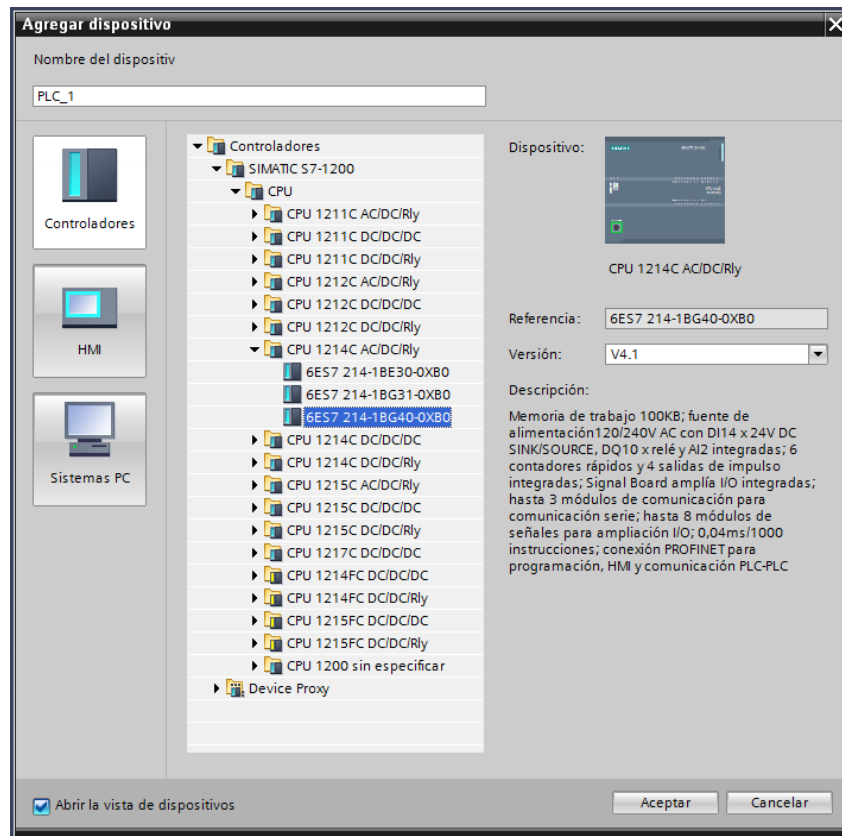


Ilustración 78: Selección del autómatas en Tia Portal

Hecho esto, queda añadir la pantalla del portátil como elemento para visualizar y crear las pantallas.

Volveremos a hacer click en “Agregar dispositivo” pero esta vez iremos a Sistemas PC → SIMATIC HMI Applications → WinCC RT Advanced y aceptaremos. Con esto añadiremos la pantalla que hemos mencionado anteriormente.

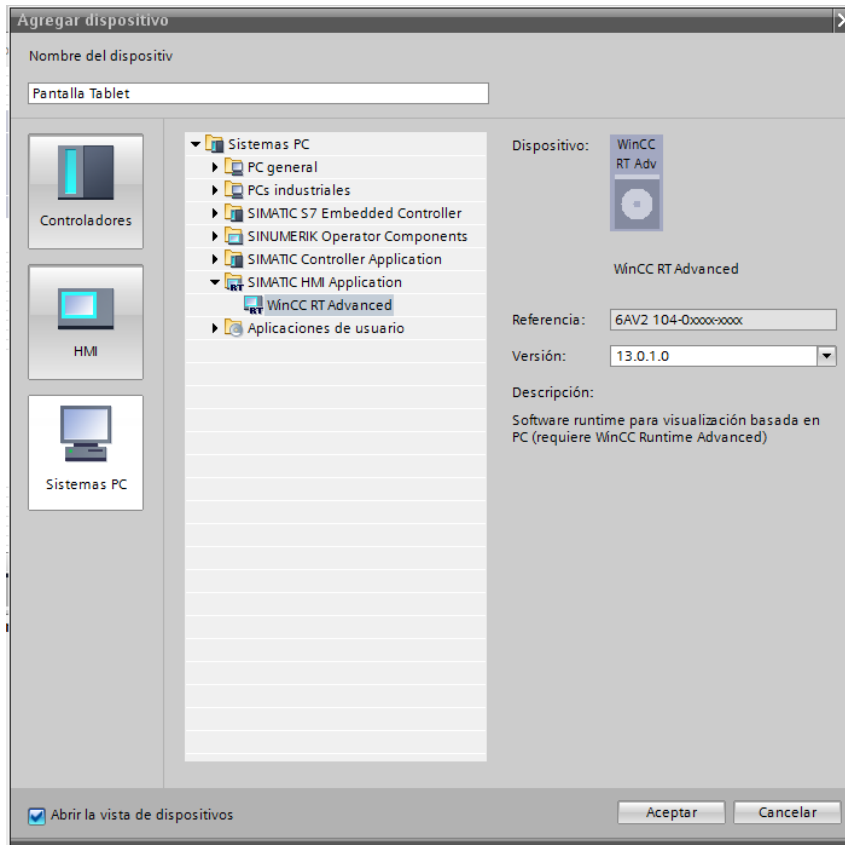


Ilustración 79: Selección de la pantalla en Tia Portal

Falta añadir una conexión de red al portátil mediante el programa. Haremos doble click en la pantalla que acabamos de crear para poder hacerle modificaciones y a continuación en la ventana denominada “Catálogo” que está ubicada a la derecha de la pantalla buscaremos y añadiremos el elemento “IE general”, que se encuentra en la ruta Módulos de comunicación → PROFINET/Ethernet.

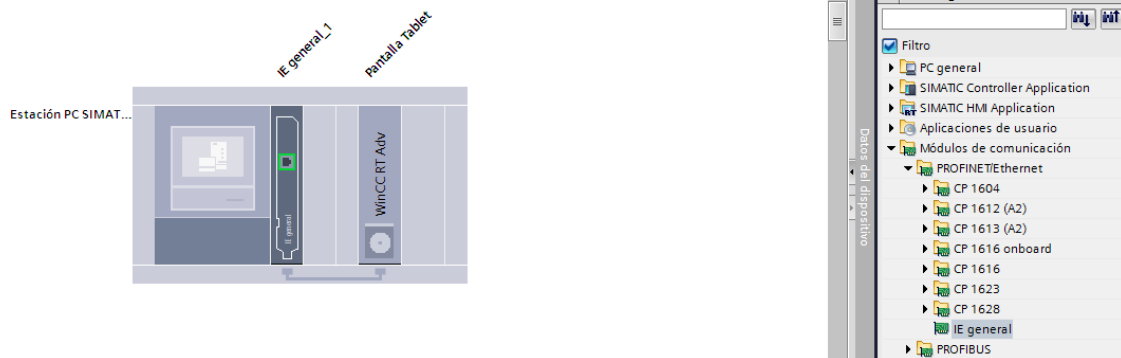


Ilustración 80: Añadido de una conexión red a la pantalla mediante Tia Portal

Ahora debemos establecer una conexión entre el autómata y el portátil. Para ello primero debemos conectar ambos físicamente, por lo que será necesario conectar mediante un cable Ethernet los puertos Ethernet que ambos tienen. Después tendremos que unir ambos dispositivos a través de la pestaña “Dispositivos y redes” y configurar sus direcciones Ip.



Ilustración 81: Conexión física entre el automático y el portátil

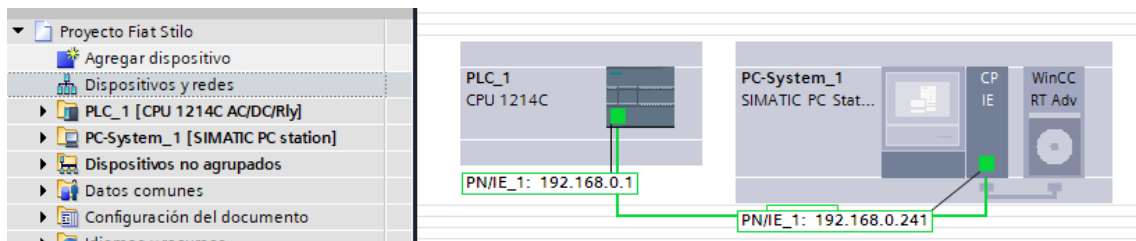


Ilustración 82: Conexión entre automático y portátil mediante Tia Portal

Cómo podemos ver en la imagen, hemos unido el automático y la pantalla haciendo click en un recuadro verde del automático y arrastrando hasta el recuadro verde de la pantalla. Como resultado de esta acción se ha creado una red entre ambos dispositivos y se les ha asignado una dirección Ip a cada uno de ellos. Esta dirección Ip siempre se puede cambiar, pero hay que asegurarse de que estas no se repitan y de que están seleccionada correctamente.

En caso de querer cambiar la dirección Ip de cualquiera de los dispositivos, haremos click en uno de ellos y en una de las pestañas denominada "General" que se encuentra en la parte inferior de la pantalla. Aquí seguiremos la ruta Interfaz PROFINET [X1] → Direcciones Ethernet → Protocolo Ip, siendo este apartado donde podremos modificar la dirección Ip.

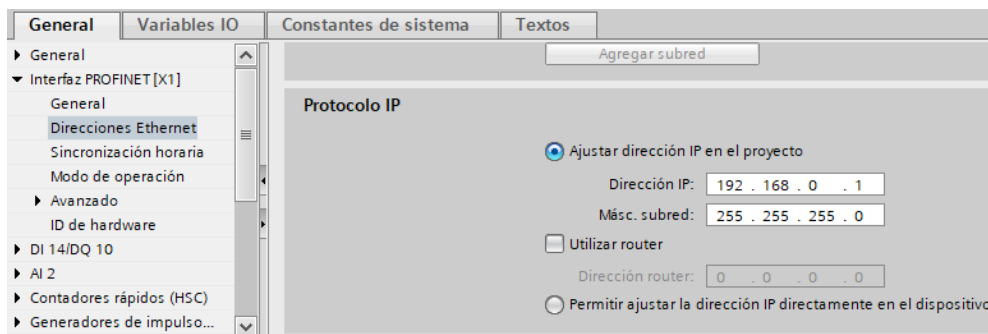


Ilustración 83: Configuración de dirección Ip mediante Tia Portal

Para finalizar este apartado queda realizar las comprobaciones pertinentes para saber si todo está correctamente conectado y configurado. Para ello trataremos de cargar el programa tal cual lo tenemos al autómata, que aunque el programa esté vacío se transferirá si todo es correcto.

Para cargar el programa en el autómata, tenemos que seleccionar el autómata y hacer click en el botón “Cargar en dispositivo” que está ubicado en la parte superior de la pantalla.

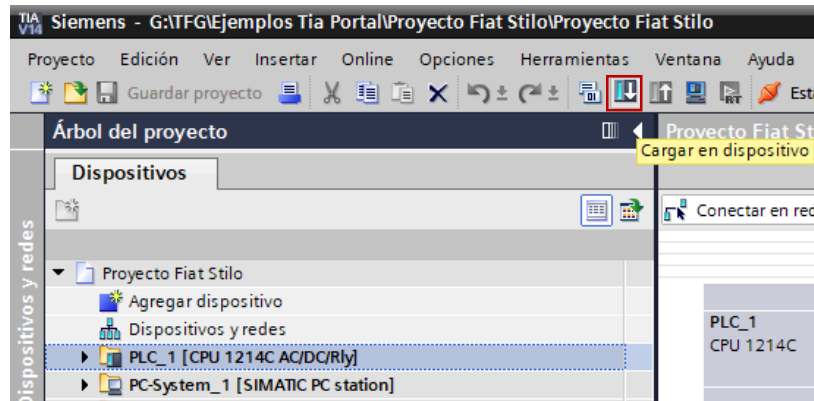


Ilustración 84: Cargado del programa al autómata mediante Tia Portal

Nos aparecerá a continuación una ventana cuya función es la de localizar al autómata. Seleccionaremos la tarjeta de red que emplea el portátil para comunicarse con el autómata y haremos click en el botón “Iniciar búsqueda” Y esperaremos hasta que se localice nuestro autómata.

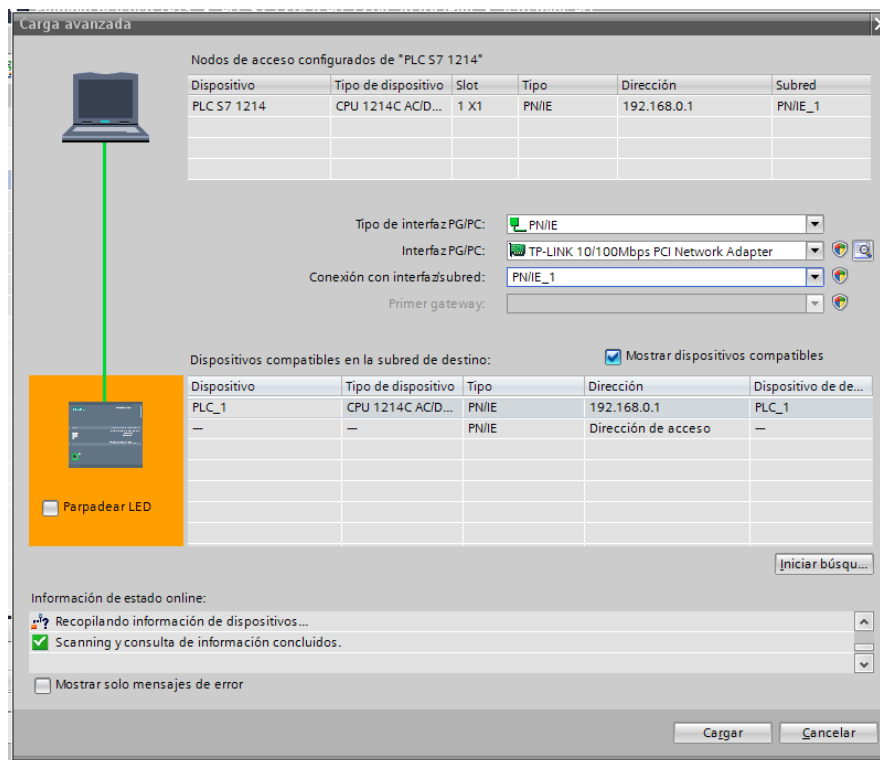


Ilustración 85: Búsqueda del autómata mediante Tia Portal

Como vemos en la imagen, podemos tener varios autómatas conectados a esa red. Podremos diferenciarlos en caso de tener más de uno mediante su dirección Ip o

mediante la opción “Parpadear LED”, que hará que los leds de comunicación del autómatas seleccionado parpadeen sin cesar. Dicho esto seleccionaremos nuestro autómatas y haremos click en el botón “Cargar”

Hecho esto aparecerá una última ventana, donde se nos muestra cualquier posible fallo o si ya existe algún otro programa en el autómatas.

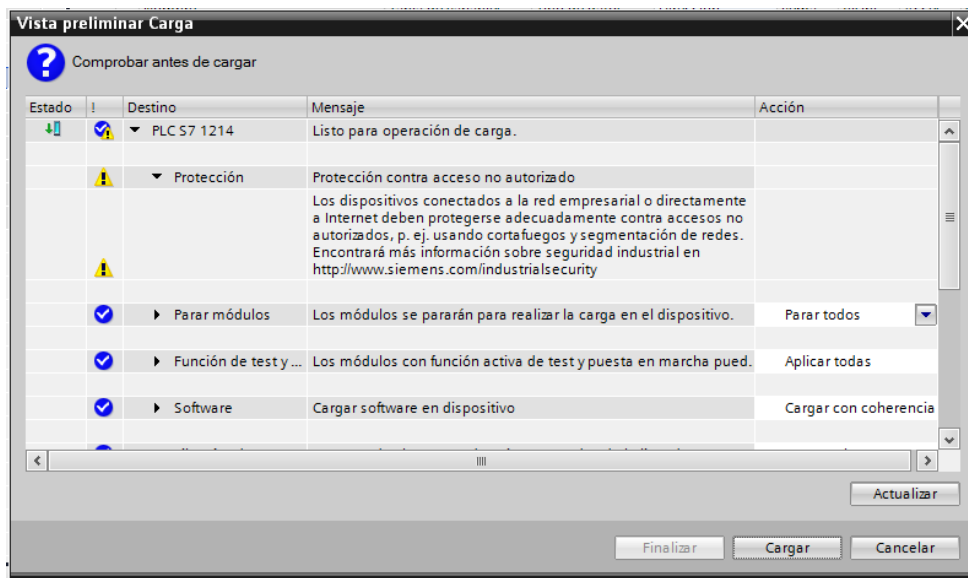


Ilustración 86: Carga de programa en el autómatas

En nuestro caso nos han aparecido una advertencia relacionada con la seguridad, que no debe de suponer ningún problema. Hacemos click en “Cargar”, esperamos a que se cargue el programa y acto seguido hacemos click en “Finalizar”, cerrando así la ventana y habiendo cargado nuestro programa en el autómatas.

Si se han seguido las instrucciones que hemos dado en este apartado o en el apartado 4.4. Configuración del autómatas, no debería de haber ningún problema a la hora de localizar el autómatas y cargar el programa en este.

Efectuadas las comprobaciones sobre el conexionado y funcionamiento del autómatas podemos continuar con este ejemplo y comenzaremos a diseñar el programa que va a ejecutar el autómatas y mostrar la HMI.

8.3. Funciones de nuestro ejemplo

Ahora vamos a explicar todas las funciones o características que nos permite hacer nuestro ejemplo. Dado que es un ejemplo vamos a simular algunas de las funciones principales de nuestro vehículo. Con este ejemplo pretendemos mostrar todas las posibilidades que tiene nuestro sistema y que no se limitan únicamente a las mostradas en este apartado.

A continuación mostraremos una a una las 3 funciones que tiene nuestro ejemplo, mostrando primero el flujograma de cada función y después una descripción de la función en cuestión. Podemos encontrar todos los bloques de nuestro ejemplo en el Documento V: Anexos.

Intermitentes

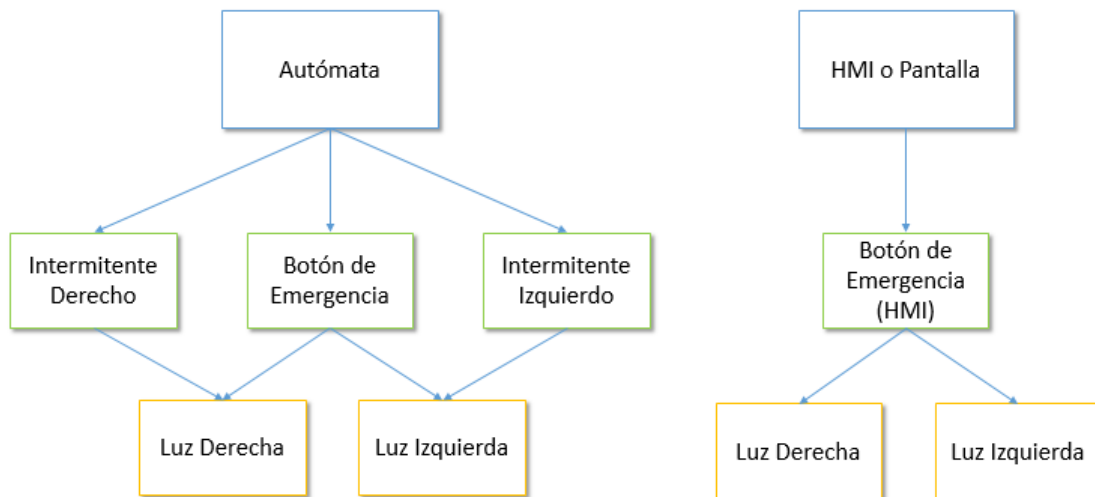


Ilustración 87: Flujograma de la función Intermitentes

Mediante esta función pretendemos emular tanto los intermitentes, al igual que las luces de emergencia.

Cada intermitente tiene asignada una entrada física (palancas del autómata), una salida física (luces del autómata) y una salida por pantalla. En cambio las luces de emergencia tienen entrada tanto física como a través de la pantalla y tienen salida por pantalla y salida física activando las luces de los intermitentes.

Como podemos ver en el flujograma, a través del autómata podemos accionar las palancas de los dos intermitentes y la de las luces de emergencia. En caso de que activemos la palanca de "Intermitente Derecho" se activará la salida "Luz Derecha", en caso de activar la palanca de "Intermitente Izquierdo" se activará la salida "Luz Izquierda", y en caso que activemos la palanca de "Botón de Emergencia" se activarán las salidas "Luz Izquierda" y "Luz Derecha".

También podemos ver que si a través de la pantalla pulsamos el botón "Botón de Emergencia HMI" también se activarán las salidas "Luz Izquierda" y "Luz Derecha".

Hay que mencionar que tanto "Luz Izquierda" como "Luz Derecha" al activarse aparecerán unas flechas en la pantalla indicando que los intermitentes están activos.

Con esta función conseguiremos simular un sistema de intermitencia típico de los vehículos. Podemos ver su funcionamiento tanto por las salidas físicas del autómata como por la pantalla.

Luces

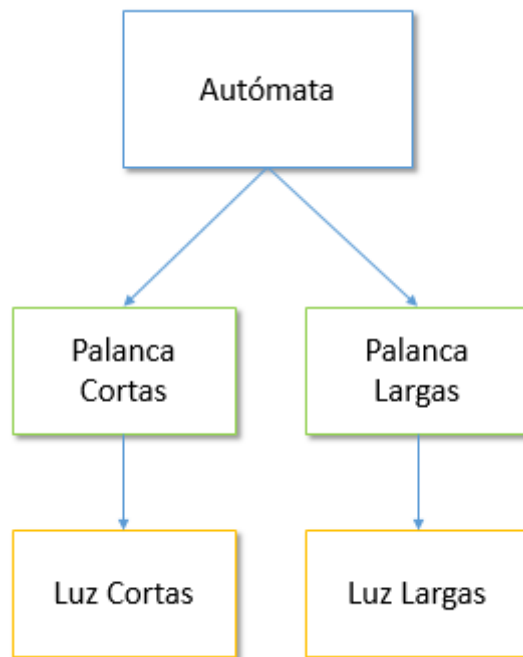


Ilustración 88: Bloque de la función Luces

Como podemos ver este flujograma es muy sencillo y únicamente controla las luces de posición y largas del ejemplo.

En esta función cuando activemos las entradas físicas del autómata denominadas “Palanca Cortas” y “Palanca Largas” se activarán las salidas “Luz Cortas” y “Luz Largas” respectivamente.

Al igual que con la función anterior, al activarse cualquiera de las salidas de esta función, se nos indicará por pantalla que están activas mediante la aparición de un icono para cada una de las salidas.

Climatización

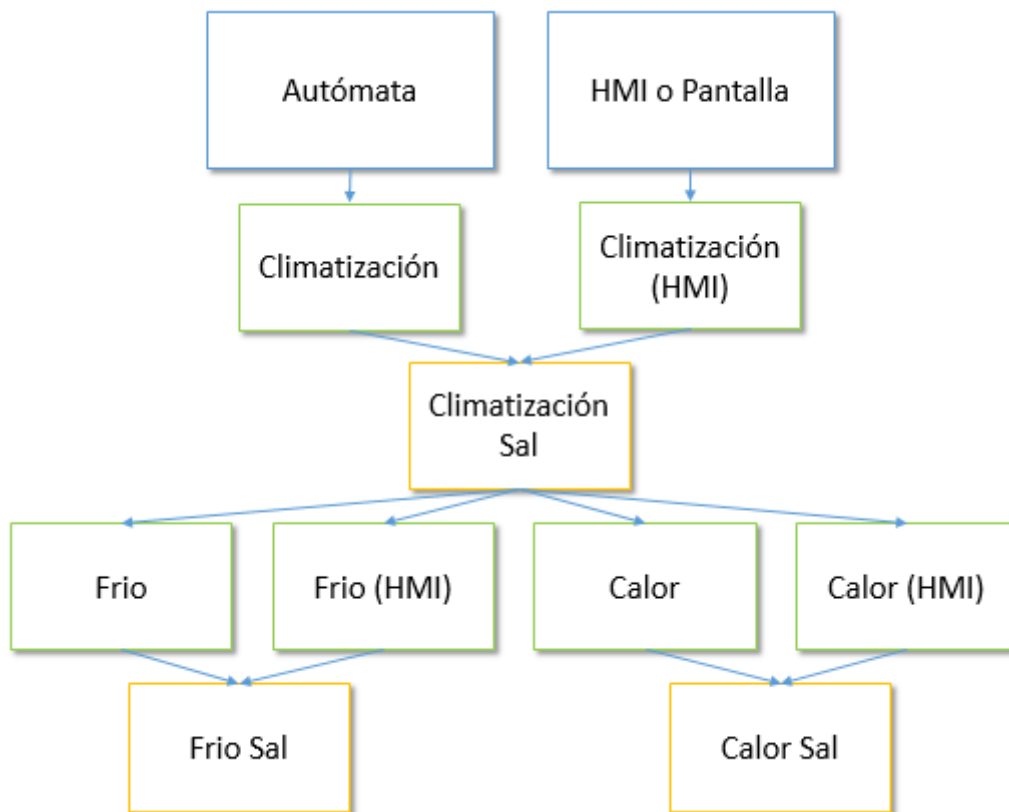


Ilustración 89: Primer Flujoograma de la función Climatización

Como podemos ver esta función es muy larga y se ha tenido que dividir en dos flujogramas.

El objetivo de esta función es el de controlar todo el sistema de climatización de nuestro vehículo, ya sea mediante las entradas físicas del autómata o mediante las entradas colocadas en la pantalla.

Primero explicaremos el primer flujograma, y posteriormente el segundo.

Mediante la entrada física “Climatización” del autómata o el botón “Climatización (HMI)” de la pantalla podremos activar la salida “Climatización Sal”. Esto se he hecho para simular el típico botón de encendido de los climatizadores, de esta forma aunque pulsemos otros botones o accionamos alguna otra entrada física, el sistema no reaccionará.

Seguidamente, si “Climatización Sal” está activada, podremos activar las entradas del autómata “Frío” y “Calor” y podremos pulsar también los botones de la pantalla “Frio (HMI)” y “Calor (HMI)”.

Finalmente si “Frío” o “Frio (HMI)” están activadas, se activará la salida física “Frio Sal”. Si “Calor” o “Calor (HMI)” están activadas, se activará la salida física “Calor Sal”. Con esto se conseguirá activar la calefacción en modo frio o calefacción.

Además de lo que hemos visto del primer flujograma, en este por motivos de seguridad si “Calor Sal” o “Frio Sal” está activada, la otra no podrá ser activada, evitando así problemas en el funcionamiento de la climatización.

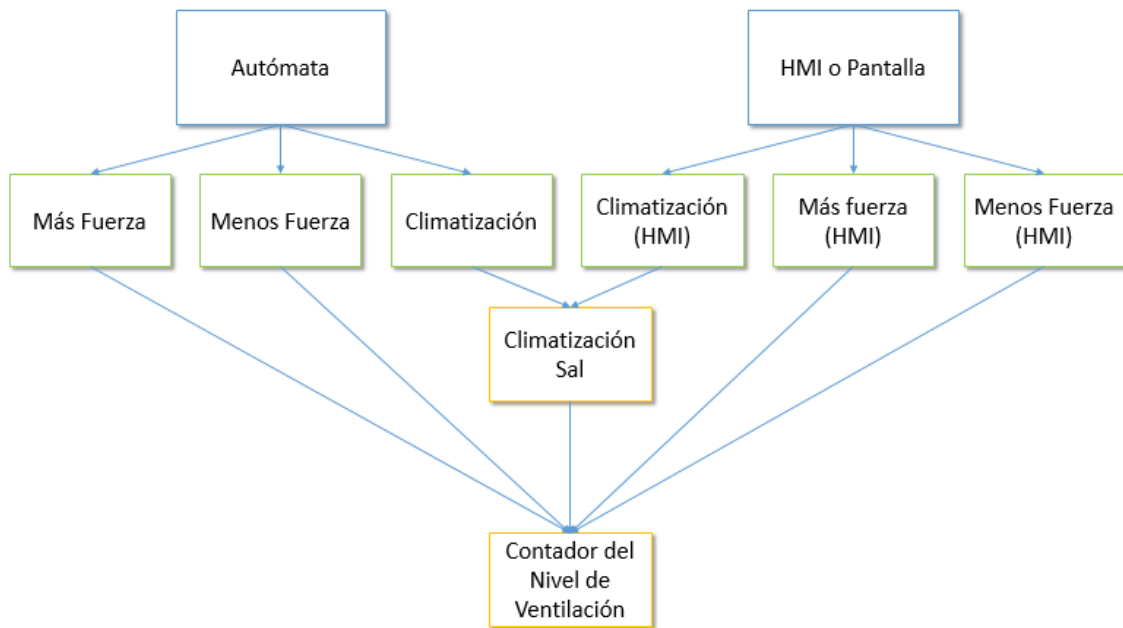


Ilustración 90: Segundo flujograma de la función Climatización

Ahora vamos a ver el segundo flujograma, cuya función es la de controlar la fuerza de la ventilación de la climatización.

No podremos controlar la fuerza de la ventilación a no ser que la salida “Climatización Sal” esté activada. En caso de que esté activa ya podemos regular la variable “Nivel de Ventilación”

El contador de la variable “Nivel de Ventilación” va de 1 a 11 (en vez de 10 por motivos de seguridad), siendo 1 el nivel mínimo de ventilación y 10 el máximo. Este nivel se puede aumentar con las variables “Más Fuerza” y “Más Fuerza (HMI)” o reducir con las variables “Menos Fuerza” y “Menos Fuerza (HMI)”, de modo que cada vez que activemos una de estas variables el contador subirá o bajará un punto. Una vez el contador ha llegado a 10, se desactivará la opción de aumentar la ventilación, pudiéndose únicamente reducir el nivel de esta. Se podrá ver el nivel de la variable “Nivel de Ventilación” en todo momento gracias a la pantalla.

Al igual que en el resto de funciones de nuestro programa, si activamos cualquiera de las salidas físicas, podremos ver su estado a través de la pantalla.

8.4. Pantallas de nuestro ejemplo

En este apartado explicaremos la función de cada pantalla, así como el funcionamiento de cada uno de los elementos de estas. Pero antes vamos a ver cómo hemos configurado la pantalla.

Para configurar el tamaño de la pantalla, así como la plantilla predeterminada y pantalla inicial, tendremos que ir a PC-System_1 [SIMATIC PC station] → HMI_RT_1 [WinCC RT Advanced] → Configuración de runtime, y en este apartado ir a la pestaña “General”. Aquí podremos configurar las opciones nombradas anteriormente, así como muchas otras.

Una vez ubicados en la pestaña “General”, seleccionaremos la resolución “1366 x 768” dado que es la resolución máxima de nuestro portátil, lo que nos garantizará una pantalla completa y sin bordes. Respecto a la pantalla inicial y plantilla predeterminada, seleccionaremos las que más nos convengan.

Llegados a este punto vamos a explicar el funcionamiento de nuestras dos pantallas, la pantalla “Principal” y la “Climatización”, así como los elementos que la componen.

Pantalla Principal

Creando esta pantalla se ha buscado simular el típico cuadro de instrumentos que nos podemos encontrar en cualquier vehículo detrás del volante.

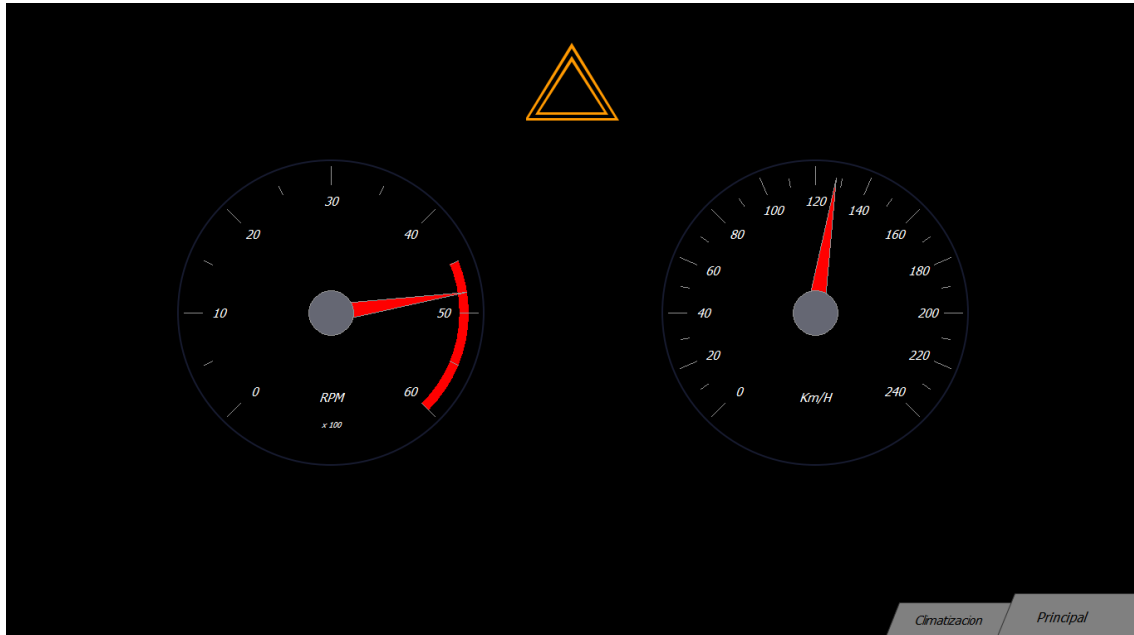


Ilustración 91: Pantalla principal de nuestro ejemplo

Esta es la pantalla principal de nuestro ejemplo, la que se muestra nada más iniciar la simulación.

Como podemos ver tenemos dos indicadores circulares que nos miden las revoluciones por minuto (izquierda) y la velocidad en kilómetros por hora (derecha). Estos indicadores van ligados a unas variables, que a su vez se controlan mediante las entradas analógicas de nuestro autómatas, que resultan ser dos potenciómetros.

Estos potenciómetros inicialmente están sin escalar y miden desde 0 hasta 23208. Por lo tanto habrá que escalarlos para hacer que midan de 0 a 60 para las revoluciones por minuto y de 0 a 240 para los kilómetros por hora.

Para escalar la variable que deseemos tenemos que acceder a ella siguiendo la ruta PC-System_1 [SIMATIC PC station] → HMI_RT_1 [WinCC RT Advanced] → Variables HMI → Mostrar todas las variables.

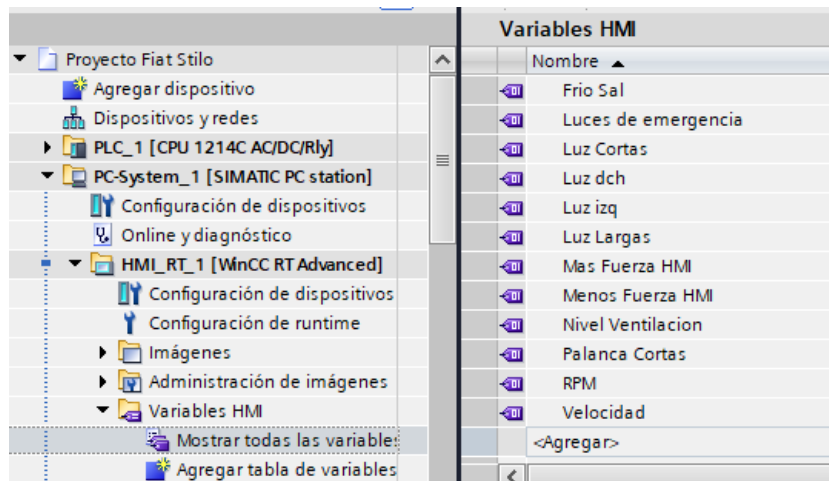


Ilustración 92: Acceder a tabla de variables de la pantalla

Ahora seleccionaremos cualquiera de las dos variables a modificar y nos iremos al apartado “escalado lineal” de la pestaña “Propiedades”.

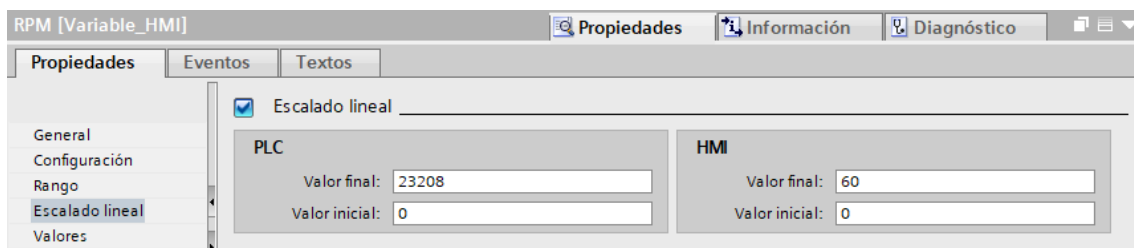


Ilustración 93: Ajustar la escala de una variable

Como podemos ver en la imagen, hemos puesto la escala real en la parte de “PLC” y luego la que queremos que se muestre por pantalla en “HMI”. De esta forma cuando lleguemos al tope de nuestro potenciómetro, habremos llegado al máximo de nuestra variable.

Dejando de lado estos dos indicadores, nos vamos a explicar el apartado de las luces, luces de emergencia e intermitentes.

Observando la ilustración 87 vemos únicamente el botón de las luces de emergencia. Esto sucede dado que no hay ninguna luz activada. En cambio sí activamos todas el cuadro quedará de la siguiente forma.

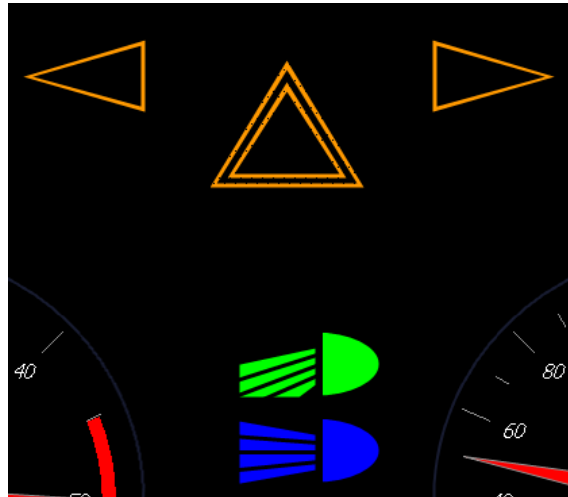


Ilustración 94: Visualización de las luces mediante la pantalla principal

Como podemos ver, tenemos un ícono para las luces cortas y otro para las largas. De forma que cuando se active cualquiera de las dos luces, aparecerá el ícono correspondiente en la pantalla.

Tenemos esta misma función para los intermitentes, donde se mostrarán unas flechas en la pantalla en función de que intermitente activemos.

También tenemos las luces de emergencia, que aparte de ser un indicador, también actúa como botón. Por este motivo podemos activar las luces de emergencia mediante las entradas físicas del autómatas o mediante el botón que se ha ubicado en la pantalla. Al activarse las luces de emergencia, tanto el triángulo central de la pantalla, como los intermitentes comenzarán a parpadear, indicando así que la salida física de estos está activada.

Por último, en la esquina inferior derecha de la pantalla nos encontramos con lo siguiente:



Ilustración 95: Indicador de pantalla y botón de cambio de pantalla

Aquí podemos ver un letrero grande que nos indica la pantalla en la que nos encontramos. Luego tenemos otro letrero más pequeño que nos indica la otra pestaña a la que podemos ir si pulsamos dicho letrero.

Gracias a esto podremos moveros por las diferentes pantallas y sabremos en que pantalla nos encontramos en todo momento.

Estos son todos los elementos que componen la pantalla principal de nuestro ejemplo. Se trata por tanto de una pantalla simple pero también limpia, que tampoco queremos que esté colapsada de información dado que será la pantalla más empleada.

Pantalla de la climatización

Esta se trata de una pantalla secundaria cuya función es la de albergar todos los controles de la climatización, además de mostrar u ofrecer todas las funciones de las que disponíamos en la pantalla principal.



Ilustración 96: Pantalla secundaria de nuestro ejemplo

Como podemos ver los indicadores circulares se han sustituido por unos más reducidos dado que había que ubicar en esta pantalla todos los indicadores y controladores del sistema de climatización. De esta forma hemos conseguido mostrar toda la información relevante del vehículo, aunque sacrificando parte de la estética de la pantalla. Por tanto esta pantalla queda destinada a un aspecto más secundario y para el control del sistema de climatización.

Como podemos ver, hemos sustituido los indicadores circulares tanto de las revoluciones por minuto, como de los kilómetros por hora.

En el caso del cuentarrevoluciones, se ha optado por uno horizontal donde se nos muestre la misma información que con el indicador circular, aunque de una forma más reducida.

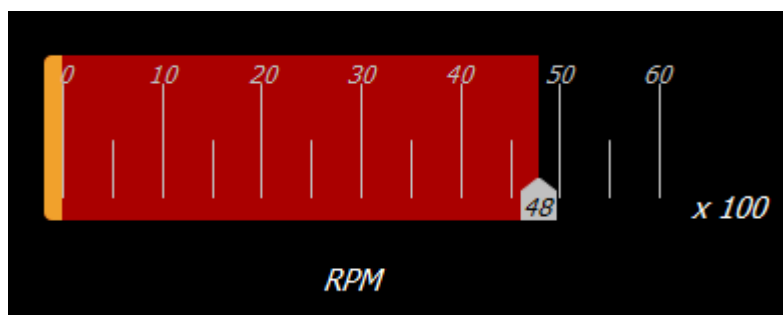


Ilustración 97: Cuentarrevoluciones horizontal

El cuentakilómetros también ha sido sustituido por un indicador más simple, donde se nos muestra únicamente la velocidad en valor numérico en vez de con aguja. De esta forma el indicador se vuelve mucho más simple y reducido.



Ilustración 98: Cuentakilómetros numérico

Ahora pasamos a ver cómo están ubicadas las luces, luces de emergencia y los intermitentes en esta pantalla.

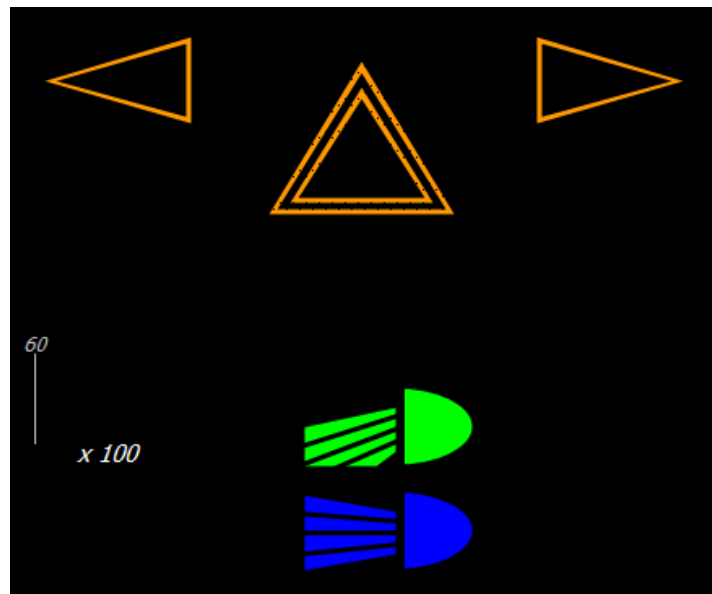


Ilustración 99: Visualización de las luces mediante la pantalla secundaria

Como podemos ver, Estos elementos de la pantalla no han variado nada en absoluto. Hemos preferido dejarlos fijos ya que nos aportan información relevante del vehículo.

Dado que son exactamente iguales a los elementos de la pantalla principal, simplemente hemos tenido que copiarlos en la pantalla de la climatización. Hecho esto todo el sistema de luces e intermitentes ya estaría preparado para su funcionamiento.

A continuación vamos a explicar cómo funciona todo el sistema de la climatización.

Comenzaremos explicando la primera parte, donde se muestra el estado de la climatización, así como el tipo de climatización.

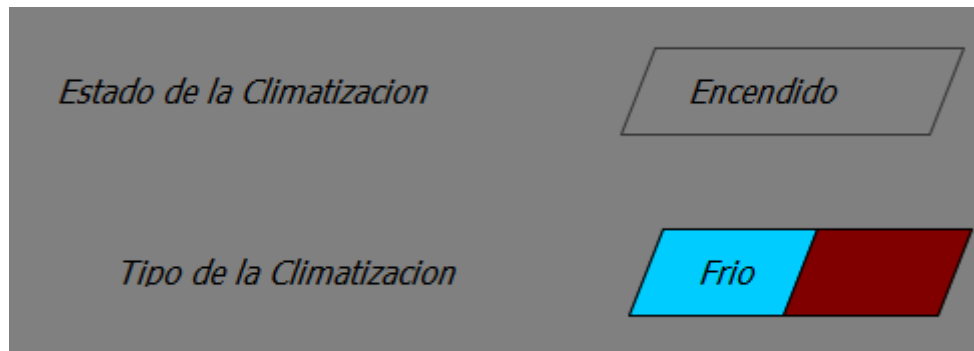


Ilustración 100: Primera parte del control de la climatización

A diferencia de la ilustración 92, en esta última se aprecia que la climatización está en funcionamiento y en modo frío.

Para encender o apagar la climatización se puede accionar la entrada física del autómata o la entrada por pantalla que hay en el letrero “Encendido”. Podremos ver si la climatización está apagada tanto por las salidas físicas del autómata, como por el letrero antes mencionado, que variará entre “Encendido” o “Apagado” dependiendo del estado de la climatización.

Para cambiar entre el tipo de climatización, tenemos dos botones en la pantalla y dos entradas físicas en el autómata. Podremos cambiar el tipo de climatización tanto con los botones de la pantalla como con las entradas físicas del autómata. Para evitar malfuncionamientos se ha impedido que ambos tipos de la climatización estén encendidos, por lo que para activar uno, primero tendremos que haber desactivado el otro. Podremos ver qué tipo de climatización está activada tanto por las salidas del autómata, como por el color y letrero de los botones de la pantalla.

En la segunda parte de la climatización tenemos el controlador de la fuerza de la climatización.

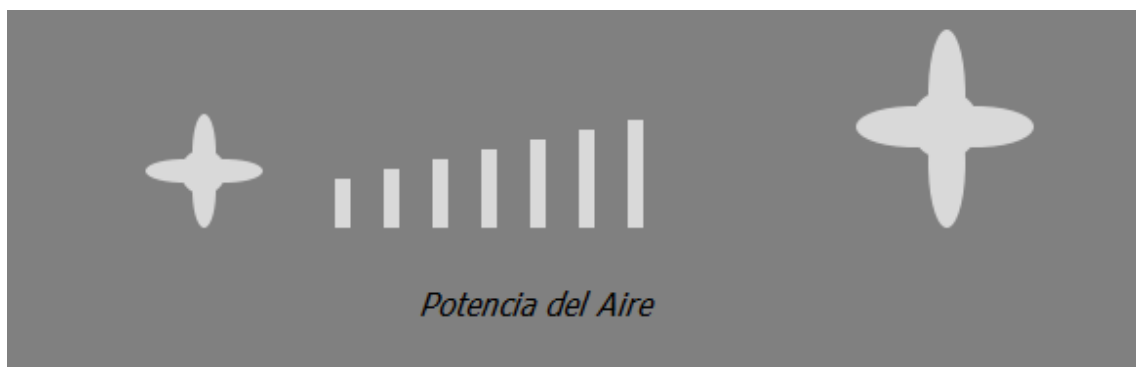


Ilustración 101: Segunda parte del control de la climatización

En la imagen podemos ver un indicador del nivel de la fuerza de la ventilación. Este nivel se podrá aumentar o disminuir mediante dos entradas físicas del autómata o mediante los botones con forma de ventilador ubicados a los lados del indicador de nivel de la pantalla. El nivel de la ventilación podrá ir de 0 a 10, y este será reconocible gracias al indicador.

Por último y al igual que en la pantalla principal, tenemos un letrero y un botón ubicados en la parte inferior izquierda de la pantalla.



Ilustración 102: Indicador de pantalla y botón de cambio de pantalla

El letrero grande nos dirá en qué pantalla nos encontramos, y el botón con letrero de al lado nos permitirá cambiar a otra pantalla.

Ya están explicados todos los elementos de esta pantalla. Como hemos podido ver a lo largo de toda la explicación, esta pantalla tiene más elementos e información que la pantalla principal, lo que la hace algo más engorrosa de entender y ver. Por este motivo tenemos como pantalla principal una pantalla que nos muestra menos información pero de forma más clara y limpia.

9. Conclusión

De este proyecto podemos sacar varias conclusiones, debido a que la mayor parte del tiempo que hemos invertido ha sido en la investigación y búsqueda de información.

Habiendo analizado la red Mini F.L.Ore.N.C.E del Fiat Stilo nos hemos dado cuenta de que todos los coches pueden tener redes distintas, a pesar de que todos ellos empleen el protocolo BUS-CAN para la comunicación en dichas redes. Esto es totalmente lógico ya que los coches en su base son iguales, pero conforme se van añadiendo sistemas auxiliares y de confort, la complejidad de las redes del vehículo va en aumento.

Otra cosa que hemos podido aprender es que el concepto de seguridad premia mucho por encima de cualquier otra característica en este tipo de proyectos. Y es esta seguridad lo que nos ha hecho decantarnos finalmente por la alternativa del autómatas, que a pesar de ser más costosa y necesitar más cableado que las otras alternativas, es imbatible en cuanto a seguridad. Y es que a 120 Km/H no te puedes permitir ni un solo cuelgue de los sistemas o interferencia en la comunicación, ya que sería fatal tanto para los ocupantes del vehículo como para los que le rodean.

También hemos visto con el ejemplo que hemos realizado en el laboratorio, las capacidades más que suficientes de nuestra alternativa para el trabajo que queremos desempeñar, y a su vez el sin fin de posibilidades que esta nos permite realizar, desde el control de la climatización del vehículo, hasta incluso poder implementar la velocidad de cruce en el vehículo.

Finalmente hay que recalcar que aunque no todos los vehículos son iguales, todos emplean la misma base, por lo que el trabajo que hemos realizado en este proyecto no sirve únicamente para el Fiat Stilo, sino que puede ser adaptado sin mucha dificultad a todo vehículo que emplee una red CAN como sistema neurálgico, que hoy en día es casi toda la totalidad de los vehículos a motor.

10. Bibliografía

- [1] *CONTROL AUTOMÁTICO: Tiempo Continuo y Tiempo Discreto*; Editorial Reverte; ISBN: 978-84-291-4751-3
- [2] *Reglamento Electrotécnico para Baja Tensión*; marcombo; ISBN: 978-84-267-2316-1
- [3] <http://www.aficionadosalamecanica.com/obd2.htm>
- [4] <https://www.adslzone.net/2017/03/24/arduino-vs-raspberry-pi-cual-es-la-mejor-placa-para-iniciarse/>
- [5] <https://www.xataka.com/makers/raspberry-pi-frente-a-arduino-quien-se-adapta-mejor-a-mi-proyecto-maker>
- [6] <http://www.vidasostenible.org/informes/historia-de-las-tecnologias-de-reduccion-de-emisiones-contaminantes-en-vehiculos/>
- [7] <http://www.areavag.com/foro/showthread.php?19076-Historia-sistemas-el%E9ctr%F3nicos-del-autom%F3vil>
- [8] http://www.adfweb.com/home/products/PROFINET_CAN.asp?frompg=matrix_12_1&loc_phy=1005545&k001=e&enk1=can%20to%20profinet&d=c&g=17882575329&gclid=Cj0KCQiAi7XQBRDnARISANeLleu9I3k2qKWuPKK0qDJnJgkkgiJeg9kEiMvMrAXmwDJ-K7zvC6MxuMaArloEALw_wcB
- [9] <https://botscience.wordpress.com/2012/06/05/historia-de-arduino-y-su-nacimiento/>
- [10] <https://arduinodehtics.weebly.com/historia.html>
- [11] <https://www.xataka.com/especiales/guia-del-arduinomaniaco-todo-lo-que-necesitas-saber-sobre-arduino>
- [12] <https://es.wikipedia.org/wiki/Arduino>
- [13] http://www.4dsystems.com.au/product/4D_Arduino_Adaptor_Shield_II/
- [14] <https://www.youtube.com/watch?v=Ly3HrE6XKNg>
- [15] http://www.4dsystems.com.au/product/gen4_uLCD_70D/
- [16] http://www.4dsystems.com.au/product/4D_Workshop_4_IDE/
- [17] <http://histinf.blogs.upv.es/2013/12/18/raspberry-pi/>
- [18] https://es.wikipedia.org/wiki/Raspberry_Pi
- [19] <https://www.youtube.com/watch?v=zU6OL3Ba6j8>
- [20] <https://www.skptechnology.co.uk/pican2-software-installation/>
- [21] <http://www.instructables.com/id/OBD-Pi/>
- [22] <https://github.com/linux-can/can-utils>
- [23] <https://elinux.org/Can-utils>
- [24] <https://www.skptechnology.co.uk/pican2-obdii-data-logger-for-raspberry-pi/>
- [25] https://en.wikipedia.org/wiki/OBD-II_PIDs
- [26] <https://projects.raspberrypi.org/en/projects/getting-started-with-guis>

Análisis y elección de diferentes sistemas para la obtención de datos de la centralita de un vehículo para su posterior muestra por pantalla

Documento III: Pliego de condiciones

Febrero de 2018

Universitat Politècnica de València
Grado en Ingeniería Eléctrica

Autor: Carlos Vicente López

Índice del Pliego de Condiciones

1.	Introducción	94
2.	Objetivo del pliego de condiciones	94
3.	Condiciones a satisfacer por los componentes	94
3.1.	Conductores eléctricos	94
3.2.	Cables Ethernet.....	95
3.3.	Transformadores.....	95
3.4.	Autómata programable	95
3.5.	Conexionado	96
3.6.	Tablet	96
4.	Condiciones facultativas.....	96
5.	Condiciones económicas	97
5.1.	Mejoras del proyecto inicial	97
5.2.	Pago de los trabajos.....	97
6.	Condiciones legales	98
6.1.	Contrato	98
6.2.	Adjudicación de la contrata	98
6.3.	Arbitraje y jurisdicción	98
6.4.	Impuestos.....	98
6.5.	Rescisión del contrato.....	98

1. Introducción

El pliego de condiciones de un proyecto se trata del documento contractual que fija las relaciones entre propiedad y contratante. Este pliego recoge todas las exigencias técnicas, económicas y legales que han de tenerse en cuenta para la ejecución, puesta en marcha y posterior mantenimiento del proyecto.

2. Objetivo del pliego de condiciones

El objetivo de este documento viene a definir los requerimientos y reglas relacionadas, económicas, legales, potestativas y técnicas del proyecto. Este documento se ajustará en los aspectos técnicos que abarcan el presente proyecto. Se efectuará una descripción de los procesos constructivos y productos manejados.

Este documento abarca cuatro tipos básicos de condiciones:

-Condiciones técnicas: Hacen referencia a los trabajos que hay que realizar, las características y calidad de los materiales, cuidados especiales y detalles concretos a tener presente durante la ejecución, y a los controles y ensayos de calidad preceptivos.

-Condiciones facultativas: Hacen referencia a los derechos y obligaciones de las partes y sus representantes en el momento de ejecutar el proyecto.

-Condiciones económicas: Hacen referencia a las garantías, la formación de precios, las formas de abono y las indemnizaciones por incumplimiento.

-Condiciones legales: Hacen referencia al perfil de contratista, la forma de adjudicación, el tipo de contrato, la obligatoriedad de suscripción de seguros de responsabilidad civil y otros asuntos relacionados.

Con todas estas condiciones dicho documento realizará una descripción detallada de la normativa legal a la que está sujeta el proyecto y la seguridad y calidad tanto del proceso de montaje como de la ejecución del mismo

3. Condiciones a satisfacer por los componentes

3.1. Conductores eléctricos

Todos los conductores eléctricos empleados en la instalación deberán de estar aislados mediante PVC para evitar cortocircuitos, evitar derivaciones a tierra y proteger a los usuarios de un contacto directo.

Estos conductores pueden ser de cobre o aluminio, siempre y cuando la sección de estos se adapte a la potencia que circule por el mismo conductor.

Los conductores del sistema que trabajen con tensiones de 24, 12 o 5 V deberán de tener secciones de 0,5 mm².

Todos los conductores eléctricos deberán de estar fijados a superficie con tal de evitar vibraciones y a su vez para evitar esfuerzos mecánicos que puedan producir deterioro de los mismos.

Todos los conductores eléctricos deberán cumplir la ITC-BT-07.

3.2. Cables Ethernet

Todos los cables tipo Ethernet destinados a la transmisión de datos deben de ser tipo par trenzado sin blindar (UTP), par trenzado apantallado (FTP), o par trenzado blindado (FTP). No se permitirá el uso de otro tipo de cables Ethernet para este proyecto.

La torsión del cable no debe superar la medida de 4 veces el diámetro del cable.

La distancia máxima entre conectores no debe exceder los 90 metros.

Todos los cables Ethernet deberán de estar fijados a superficie con tal de evitar vibraciones y a su vez para evitar esfuerzos mecánicos que puedan producir deterioro de los mismos.

Los cables Ethernet deberán seguir la normativa EIA/EIA-568B

3.3. Transformadores

Todos los transformadores empleados en este proyecto deberán de ser mínimo Transformadores de Seguridad con o sin toma de tierra. No se permite el empleo de transformadores cuyos circuitos no dispongan de asilamiento de protección.

Los transformadores deberán estar ubicado en una zona estanca y alejados de posibles fuentes de calor.

El cableado empleado por estos deberá tener una sección de 0,5 mm².

Todos los transformadores deberán de estar fijados a superficie con tal de evitar vibraciones y a su vez para evitar esfuerzos mecánicos que puedan producir deterioro de los mismos.

Los transformadores deberán seguir la normativa ITC-BT-36

3.4. Autómata programable

El autómata programable durante su servicio tendrá que estar a una temperatura entre -20 °C y 60 °C, y a una presión comprendida entre 795 y 1080 hPa.

La humedad del ambiente no podrá exceder el 95 % mientras el autómata programable esté en servicio.

Durante el transporte del autómata tendrá que estar a una temperatura entre -40 °C y 70 °C, y a una presión comprendida entre 7660 y 1080 hPa.

Durante el funcionamiento del autómata, este deberá estar fijado a pared para evitar movimientos indeseados y vibraciones extremas.

El autómata deberá de estar alejado en todo momento de agentes externos como polvo, líquidos o corrosivos.

La tensión nominal que se le debe suministrar al autómata debe de ser de 24 V, admitiendo valores mínimos de 20,4 V y valores máximos de 28,8 V.

El autómata deberá de seguir el estándar internacional IEC 61131-2 y IEC 61131-3.

3.5. Conexionado

Todas las conexiones que se hagan a través del puerto EOBD del Fiat Stilo1.9 JTD de 2002 así como la transmisión de datos que se efectuó con su red Mini Mini F.L.Ore.N.C.E, deberán de seguir los estándares ISO 9141, ISO 11898 e ISO 15765.

3.6. Tablet

La pantalla Tablet que se emplee para la visualización de la interfaz gráfica deberá de tener las siguientes características:

- Emplear el sistema operativo Windows 10
- Un mínimo de 2 GB de RAM
- Un mínimo de 16 GB de almacenamiento interno
- Pantalla con una resolución no inferior a los 1280 x 720 píxeles

4. Condiciones facultativas

El objetivo de este proyecto es estudiar las distintas posibilidades que existen para comunicarse con la red interna de un vehículo y posteriormente realizar un ensayo con la alternativa que haya sido seleccionada.

El ensayo que se ha realizado es un ejemplo de las posibilidades del sistema, siendo por tanto totalmente teórico.

En caso de querer llevar este ensayo a la práctica, requerirá el volver a estudiar el vehículo con el que trabajamos y a su vez mejorar el ensayo, con el sobrecoste que ello conlleve.

5. Condiciones económicas

5.1. Mejoras del proyecto inicial

Dado que el objetivo de este proyecto es el de analizar distintas alternativas y posteriormente crear un ensayo a modo de ejemplo, no será necesario implementar ninguna mejora.

En caso de que se realice su fabricación con el fin de comercializar o cualquier intento de mejora del mismo ensayo, se deberá consultar con el proyectista antes de ser realizada. En caso de que el proyectista estudiara dicha comercialización o mejora y viera que es legal y técnicamente válida y se quisiera por tanto llevar a cabo, debería de renegociarse las condiciones del contrato.

5.2. Pago de los trabajos

En esta sección del pliego de condiciones se establecen las condiciones a seguir por el contratista, el proyectista y el contratante en referencia a los pagos.

- Si el pago se realiza entre los 5 días naturales posteriores a la recepción definitiva del prototipo, inclusive con antelación a los mismos, el importe no sufrirá recargo alguno.
- Cuando el pago se realice entre 6 y 30 días naturales después de la recepción del producto definitivo por parte del contratante, el producto sufrirá un recargo del 2% sobre el precio inicial previsto.
- Si se realiza el pago entre los 31 y 60 días posteriores a la recepción definitiva, este sufrirá un recargo del 4% sobre el importe establecido.
- Para un aplazamiento del pago entre los 61 y 90 días tras la recepción definitiva, el producto sufrirá un recargo del 6,5% sobre el precio final.
- Si se demorase el pago entre los 91 y 300 días naturales seguidos después de la entrega del producto, el precio de este sufriría un recargo del 30% sobre el presupuesto inicial.
- En caso de que el contratante optase por pagar a plazos, debe de comunicarlo con anterioridad y establecerse previamente el número de plazos y el intervalo de tiempo entre cada pago fraccionado, sufriendo cada plazo un recargo en función del tiempo transcurrido y de acuerdo con los puntos anteriores.
- En caso del incumplimiento del pago en el término de tiempo escogido por parte del cliente, el fabricante tendrá derecho a demandarle ante los tribunales.

6. Condiciones legales

6.1. Contrato

En el contrato tiene que quedar claramente expresado el precio inicial por preparación de una serie y el coste de unidad del producto. En caso de que procediera, a estas tarifas se les podría aplicar recargos establecidos en los apartados anteriores de este mismo Pliego de Condiciones. El contrato debe realizarse siempre por escrito y cumpliendo todos los requisitos legales.

6.2. Adjudicación de la contrata

La empresa que realice la adquisición del producto, deberá ser la encargada de escoger que empresa se encarga de los trabajos de producción y compra de los elementos que componen el prototipo, de su instalación y por último de su programación.

6.3. Arbitraje y jurisdicción

La dirección facultativa se encargará de arbitrar las distintas partes en diferencia en caso de producirse. En caso de no llegar a un acuerdo, se reunirán un técnico por cada una de las partes, de forma que estos se encargarán de llegar a un entendimiento. Si a pesar de esto el desacuerdo prevalece, se recurrirá a la toma de control por parte de los tribunales ordinarios de justicia de la ciudad que se estime conveniente.

6.4. Impuestos

Se demandará a la contrata que esté al corriente de los pagos de impuestos, tasas y contribuciones necesarias para el desarrollo de sus actividades empresariales.

Para la comercialización del producto se añadirá un impuesto sobre el precio del mismo correspondiente al impuesto sobre el valor añadido (IVA) que está estipulado actualmente en un 21 %.

En caso de que el IVA sea modificado en un futuro por el gobierno del estado, se deberá ajustar este impuesto con tal de cumplir con esa modificación.

6.5. Rescisión del contrato

Debido a una demora excesiva en la entrega del producto sin previo aviso del contratista al contratante. Dejado atrás el plazo de entrega voluntaria, el contratante podrá o no, según si así lo desea, hacer válido el contrato habiéndole llegado el producto con tal retraso.

Análisis y elección de diferentes sistemas para la obtención de datos de la centralita de un vehículo para su posterior muestra por pantalla

Documento IV: Presupuesto

Febrero de 2018

Universitat Politècnica de València
Grado en Ingeniería Eléctrica

Autor: Carlos Vicente López

Índice del Presupuesto

1.	Presupuesto de compra de material	101
2.	Presupuesto de la mano de obra	101
3.	Presupuesto de licencias o software	102
4.	Coste de herramientas empleadas.....	102
5.	Coste neto.....	103
6.	Margen de beneficio.....	103
7.	Coste final del proyecto.....	104

1. Presupuesto de compra de material

En el siguiente presupuesto se listarán todos los componentes y materiales que hemos empleado para elaborar la alternativa del autómata programable.

Componente	Distribuidor	Precio (€)
Siemens 6ES7212-1HE40-0XB0	Contaval-online	212,8
Conector Macho EOBD	Amazon	2,83
Transformador 12 V a 24 V	Amazon	10,99
Convertidor HD67606-A1	ADFweb	358,15
Tablet Teclast X80 Pro	Amazon	72,99
Transformador 12 V a 5 V	Amazon	7,01
Adaptador Micro-usb a Ethernet	Amazon	9,99
Ladrón de puertos Micro-usb	Amazon	8,55
Switch Tp-link tl-sf1005d	Amazon	10,16
Precio total (€)		693,47

Tabla 11: Costes de la alternativa escogida, autómata programable

2. Presupuesto de la mano de obra

En el siguiente presupuesto se listarán todas las labores realizadas durante la elaboración del TFG, así como también se mostrarán las horas dedicadas a cada labor y el precio de dichas horas. El coste de las horas se ha establecido según los valores actuales del mercado.

Concepto	Horas totales	Coste de la hora (€)	Coste (€)
Análisis de los diferentes sistemas	180	16	2880
Redacción del TFG	110	18	1980
Realización del ejemplo práctico	80	21	1680
Coste total (€)			6540

Tabla 12: Coste de las horas empleadas en el proyecto

3. Presupuesto de licencias o software

En el siguiente presupuesto se listarán todas las licencias o software que han sido necesarias durante todo el proyecto. Se mencionarán también tanto el coste de la licencia o software por año, así como el tiempo que se ha estado empleando dicha licencia o software.

Concepto	Precio por año (€)	Días empleados	Coste (€)
Tia Portal Advanced	1700	80	372,60
Microsoft Office Professional Plus 2013	59,9	115	18,87
Coste Total (€)			372,60

Tabla 13: Coste de licencias o software contratados

4. Coste de herramientas empleadas

En el siguiente presupuesto se listarán todas las herramientas e instrumentos que hemos empleado para la realización de este proyecto. Pa este presupuesto tendremos en cuenta, el tiempo que hemos empleado dicho instrumento, su vida útil y su precio de compra.

Concepto	Coste (€)	Días empleados	Vida útil (años)	Coste (€)
Ordenador portátil	800	120	6	43,84
Coste Total (€)				6,00

Tabla 14: Coste de herramientas empleadas

5. Coste neto

En el siguiente presupuesto veremos cuantos gastos le ha supuesto a la empresa este proyecto. Tendremos en cuenta para este presupuesto las licencias empleadas, los materiales necesarios, la mano de obra invertida y las herramientas utilizadas.

Concepto	Coste (€)
Materiales	693,47
Horas invertidas	6540
Licencias o software	372,60
Herramientas empleadas	6,00
Coste neto (€)	7612,07

Tabla 15: Coste neto

6. Margen de beneficio

El objetivo de toda empresa es conseguir un beneficio debido a una labor realizada o un producto vendido. Debido a esto tenemos que calcular un margen de beneficios para la empresa. Observando el estado actual del mercado, estimamos en un 21% el margen de beneficio de la empresa. Este porcentaje se debe de aplicar sobre el coste neto del proyecto.

Concepto	Coste (€)
Coste neto	7612,07
Margen de beneficio (21%)	1598,54

Tabla 16: Margen de beneficio

7. Coste final del proyecto

En el siguiente presupuesto se engloban todos los gastos que ha supuesto este proyecto. Se tienen en cuenta por tanto el coste neto del proyecto, su beneficio industrial y el 21% de IVA actual que se aplica a ambos por igual.

Concepto	Coste (€)
Coste neto	7612,07
Margen de beneficio	1598,54
IVA (21%)	1934,23
Coste total (€)	11144,84

Tabla 17: Coste total del proyecto

El coste final del proyecto asciende a un total de **ONCE MIL CIENTO CUARENTA Y CUATRO EUROS CON OCHENTA Y CUATRO CENTIMOS.**

Análisis y elección de diferentes sistemas para la obtención de datos de la centralita de un vehículo para su posterior muestra por pantalla

Documento V: Anexos

Febrero de 2018

Universitat Politècnica de València
Grado en Ingeniería Eléctrica

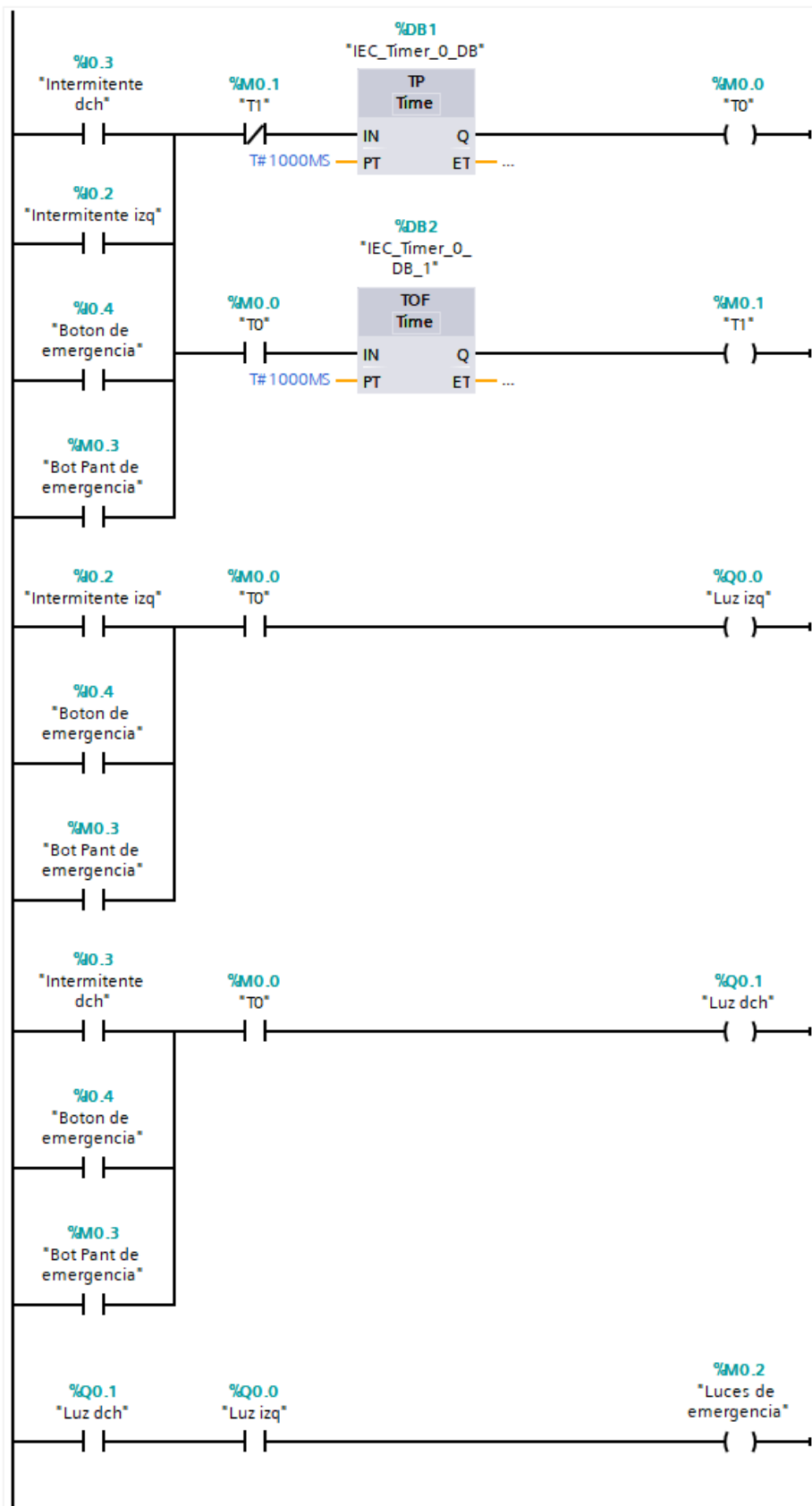
Autor: Carlos Vicente López

Índice del Anexo

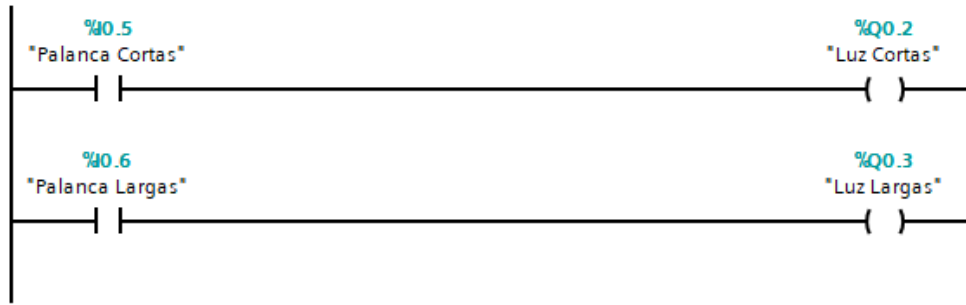
1.	Bloques de función de la aplicación implementada.....	107
1.1.	Bloque de la función “Intermitentes”	107
1.2.	Bloque de la función “Luces”	108
1.3.	Bloque de la función “Climatización”	109
2.	Esquemas.....	110
2.1.	Esquema eléctrico.....	110
2.2.	Esquema de red	111

1. Bloques de función de la aplicación implementada

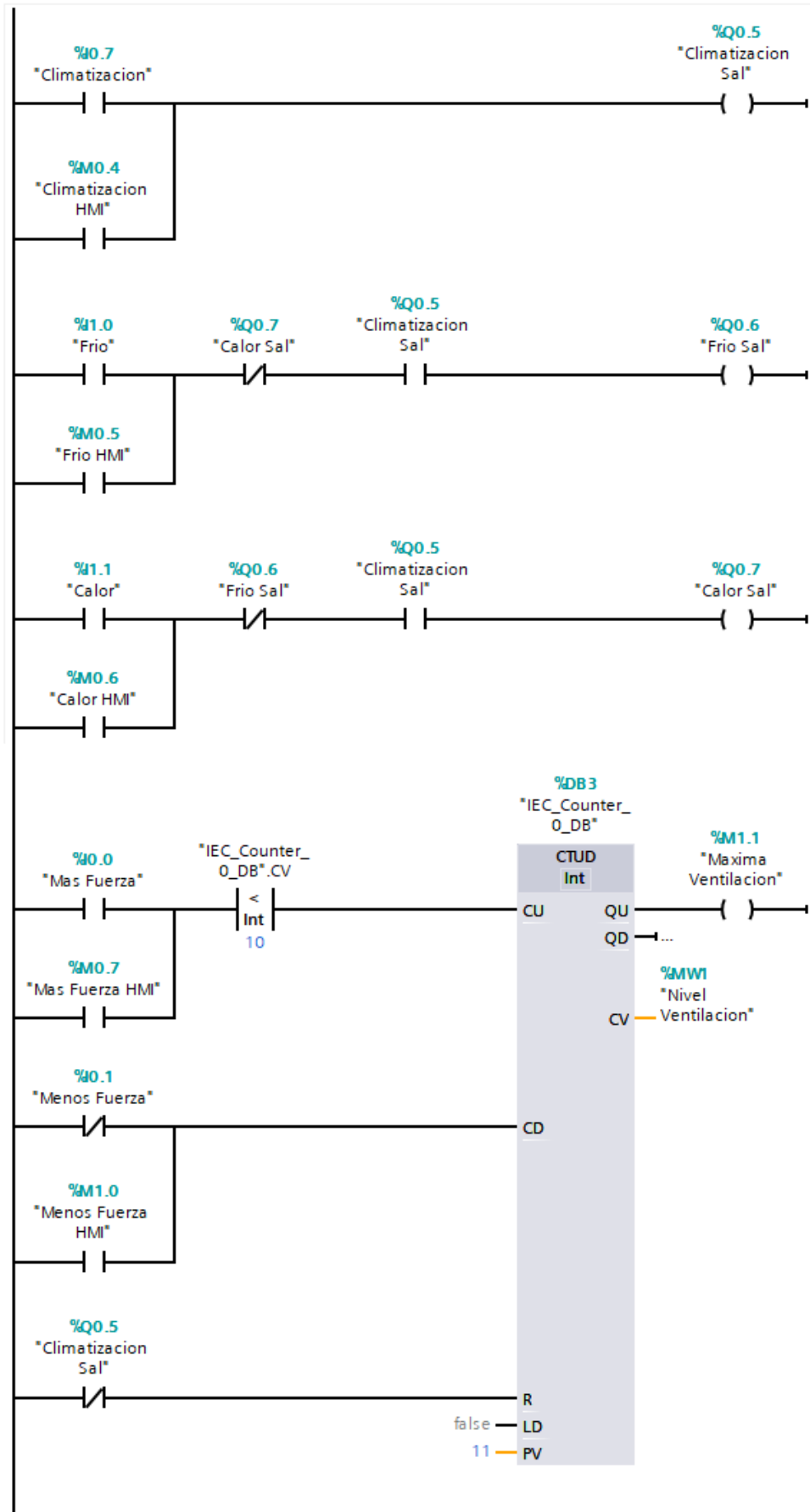
1.1. Bloque de la función "Intermitentes"



1.2. Bloque de la función "Luces"

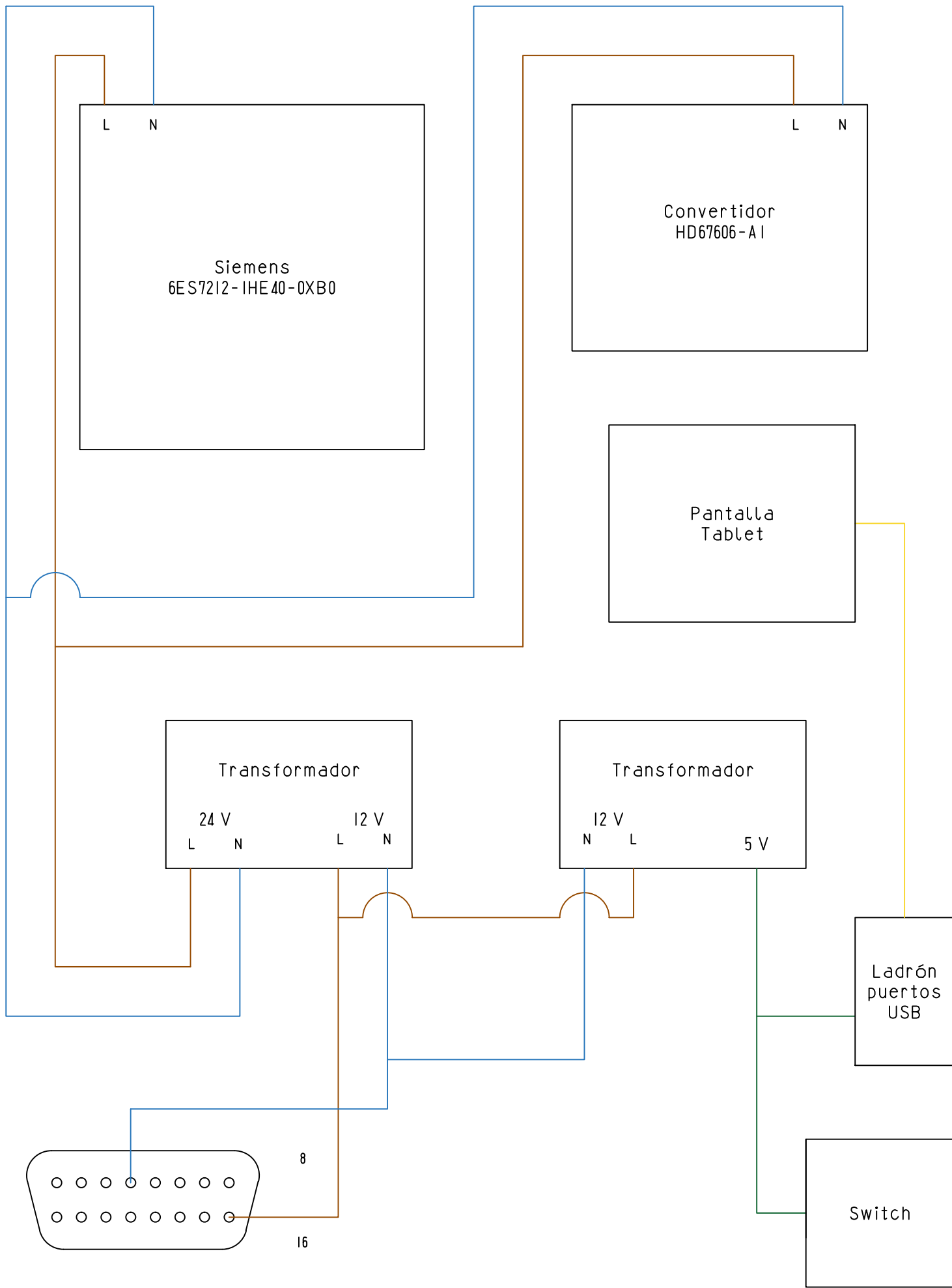


1.3. Bloque de la función "Climatización"







2. Esquemas

2.1. Esquema eléctrico

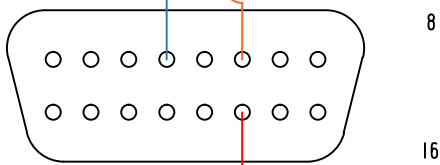
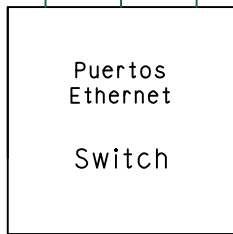
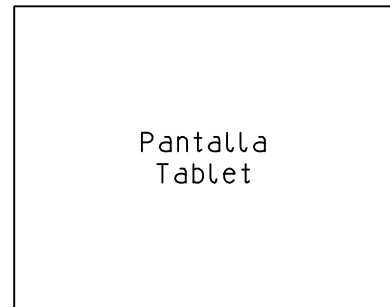
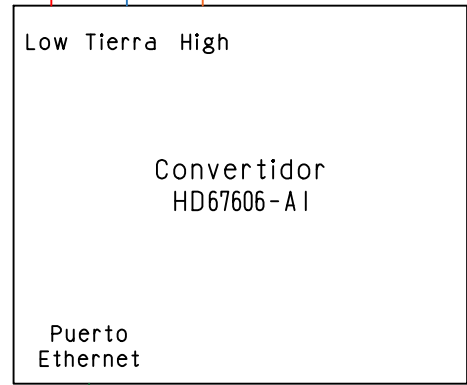


leyenda de cableado:

			
Fase	Neutro	USB	Micro USB

Universidad: Universitat Politècnica de València	Alternativa: Autómata
Departamento: Departamento de Ingeniería Eléctrica	Plano: Esquema Eléctrico
Autor: Carlos Vicente López	Fecha: Febrero de 2018

2.2. Esquema de red



leyenda de cableado:

CAN High	CAN Low	Neutro	Ethernet	Micro USB

Universidad: Universitat Politècnica de València	Alternativa: Autómata
Departamento: Departamento de Ingeniería Eléctrica	Plano: Esquema de Red
Autor: Carlos Vicente López	Fecha: Febrero de 2018