



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DIVISIÓN DE INGENIERÍA MECÁNICA E INDUSTRIAL

DEPARTAMENTO DE INGENIERÍA MECATRÓNICA

**PANTALLA *OLED* PARA INFORMACIÓN DEL TABLERO DE
INSTRUMENTOS EN VEHÍCULOS**

TESIS PROFESIONAL

PARA OBTENER EL TÍTULO DE:

INGENIERO MECATRÓNICO

PRESENTA

REYES GODÍNEZ JUAN CARLOS

DIRECTOR DE TESIS:

M. I. SERAFÍN CASTAÑEDA CEDEÑO



MÉXICO D.F.

2012



Agradecimientos

Lo primero es agradecerle a la vida por permitirme vivir esta gran aventura de estudiar ingeniería mecatrónica que hoy culmina, pero da inicio a un nuevo viaje, él de ejercer con orgullo la profesión de ingeniero.

Quiero agradecer sobre todo a mis papás los más grandes maestros que he tenido en la vida, siendo ellos los que me ensañaron a respetar y amar a mis seres queridos, a superarme día a día y a ganarme todo con mi trabajo. Su amor, sus consejos y su apoyo forjaron al hombre que hoy en día soy, no me alcanzará la vida para agradecerles los amo.

A mis hermanas que han sido en todo momento un ejemplo para mí, Yuli tú siempre me cuidaste desde el primer día que entre a la escuela, Dianita en ningún momento me has dejado solo, también mi cuñado Irvin que siempre fue un ejemplo como ingeniero apoyándome en todo desde que decidí estudiar ingeniería, a los tres los quiero mucho y los admiro.

Le agradezco a mi hermano Teddy que siempre me ha acompañado en todas mis locuras y sin importar lo que yo haga, él siempre estará para tenderme una mano. A todos mis seres queridos tíos, primos, sobrinos, amistades, mascotas, etc., cada uno de ustedes han jugado un papel importante en mi vida A mis amigos de la facultad y de la sala de proyectos he aprendido mucho de ustedes.

Me siento orgulloso de haber estudiado en la Universidad Nacional Autónoma de México que me acogió desde CCH para posteriormente darme paso a mi alma mater, la facultad de ingeniería donde los profesores me enseñaron los conocimientos necesarios para ejercer esta honrosa profesión. En especial al maestro que admiro y respeto Serafín Castañeda que confió en mí, guiándome en la aventura de la tesis.

A Mar mi gran compañera de toda esta travesía, siempre me has apoyado en todo momento, con tus palabras de aliento, tu mirada sincera, tu chispa ocurrente y sobre todo con tu amor, me has dado fuerza para seguir adelante, es un honor compartir mi vida junto a ti, te amo.

Me siento muy orgulloso de todo este trabajo que culmina con mi titulación como ingeniero mecatrónico es uno de los logros mas importantes de mi vida, pero ustedes son mi más grande orgullo, porque sé que sin todo su cariño esto no habría sido posible, solo me resta entregarles mi eterna gratitud.

El logro es de ustedes ¡Los Amo... Gracias!

**Juan Carlos Reyes Godínez
2012**



Índice

Introducción 2

Antecedentes 4

1.1 La industria automovilística..... 5

1.2 El automóvil del futuro 6

 1.2.1 Módulos de innovación de confort y seguridad..... 6

1.3 Sistema Head-Up Display (HUD)..... 8

 1.3.1 Definición del Sistema HUD 8

 1.3.2 Principio de Funcionamiento 9

 1.3.3 Tipos de sistemas HUD..... 9

 1.3.4 Partes que componen el sistema.....10

 1.3.5 Generación de la imagen.....10

 1.3.6 HUD como equipamiento de fábrica10

 1.3.7 Propuestas de futuro en HUD.....11

1.4 Pantalla OLED (Organic Light-Emitting Diode)13

 1.4.1 Introducción13

 1.4.2 Historia15

 1.4.3 Componentes de los OLEDs16

 1.4.4 Descripción de los componentes de un OLED.....16

 1.4.5 Aplicaciones de las pantallas OLEDs.....17

1.5 Planteamiento del problema18

 1.5.1 Necesidad.....18

 1.5.2 Justificación18

Diseño conceptual.....19

2.1 Objetivo20

 2.1.1 Hipótesis.....20

 2.1.2 Alcances.....20

2.2 Requerimientos y Necesidades.....20

2.3 Funciones del sistema21

 2.3.1 Descripción de las funciones.....22

2.4 Matriz de conceptos22

 2.4.1 Matriz de selección23



2.5	Arquitectura de sistema	24
2.5.1	Etapa de adquisición de la información:.....	25
2.5.2	Interfaz de comunicación vehículo - procesamiento:.....	25
2.5.3	Etapa de procesamiento de información	25
2.5.4	Interfaz de comunicación procesamiento – pantalla	26
2.5.5	Visualización de la información en la pantalla.....	27
2.5.6	Montaje del sistema en el Automóvil	27
	Desarrollo del modelo funcional	28
3.1	Preámbulo	29
3.2	Adquisición de Información	29
3.2.1	Elementos para mostrar en el parabrisas.	29
3.2.2	Funcionamiento del escáner ELM327	31
3.3	Interfaz de comunicación Vehículo - Procesamiento	32
3.3.1	Comunicación bluetooth	33
3.4	Módulo de Procesamiento de información	36
3.4.1	1° Parte del algoritmo.....	39
3.4.2	2° Parte del algoritmo.....	40
3.4.3	3° Parte del algoritmo.....	41
3.4.4	4° Parte del algoritmo.....	42
3.5	Interfaz de comunicación Procesamiento - Pantalla	43
3.5.1	Controlador gráfico GOLDELOX	44
3.6	Visualización de la información en la pantalla	47
3.6.1	Comandos de control para pantalla.....	48
3.6.2	Memoria uSD en pantalla OLED.....	49
3.6.3	Representación grafica del tablero de instrumentos para vehículo en pantalla OLED.....	51
3.7	Montaje del sistema en el automóvil	53
3.7.1	Ubicación de la pantalla OLED en el automóvil.....	53
3.7.2	Conexiones del modelo funcional	54
3.7.3	Diseño final del modelo funcional CAD	55
	Pruebas y resultados	56
4.1	Pruebas en laboratorio (PC)	57
4.1.1	Simular el comportamiento de un automóvil con la PC.....	57
4.1.2	Descripción de la prueba en laboratorio (PC)	58
4.1.3	Resultados de las pruebas en laboratorio:	59



4.2 Pruebas en campo (Automóvil) 60

 4.2.1 Automóviles a utilizar para pruebas del sistema 60

 4.2.2 Descripción de la prueba en campo (Automóvil) 60

 4.2.3 Resultados de las pruebas en campo: 61

Conclusiones 64

Trabajo a futuro 66

Bibliografía y referencias 67

Apéndice A 70

Apéndice B 78

Apéndice C 83

Apéndice D 89

Apéndice E 101

Índice de ilustraciones

Ilustración 1.1 Logotipos industria automotriz, fuente www.autox4.com 5

Ilustración 1.2 Sistema Hella de Iluminación en curva, fuente www.hella-shop.com.mx 7

Ilustración 1.3 Sistema City Safety, fuente www.volvocars.com/us 7

Ilustración 1.4 Sistema ADT Nissan, fuente www.nissan-global.com 8

Ilustración 1.5 Sistema Head-Up Display, fuente aerotecnologia.blogspot.mx 8

Ilustración 1.6 HUD Toyota, fuente www.toyota.com.au/prius 9

Ilustración 1.7 HUD BMW, fuente www.bmw.com 10

Ilustración 1.8 HUD Peugeot, fuente www.peugeotforums.com 11

Ilustración 1.9 Realidad Aumentada por General Motors, fuente www.autopasion.com 11

Ilustración 1.10 Sistema Aida 2.0, fuente www.motorpasionfuturo.com 12

Ilustración 1.11 Organic Light Emitting Diode (OLED), fuente www.bombillasoled.com 14

Ilustración 1.12 Primeros OLED, fuente comunicantropo.com/ 15

Ilustración 1.13 Estructura de una pantalla OLED, fuente www.hormigasolar.com 16

Ilustración 1.14 Laptop Samsung 14" OLED, cortesía de www.gadgetreview.com 17

Ilustración 1.15 TV OLED LG, fuente www.xataka.com 17

Ilustración 3.1 Elementos del tablero, fuente www.themotorreport.com.au 30

Ilustración 3.2 Escáner ELM327, fuente www.diagnosticoautomotor.com 30

Ilustración 3.3 Antena Bluetooth BlueSMiRF, fuente www.sparkfun.com 34

Ilustración 3.4 Placa Arduino Mega 2560, fuente www.arduino.cc/es 37

Ilustración 3.5 Pantalla OLED 4D Systems, fuente www.4dsystems.com.au 44

Ilustración 3.6 uUSB-MB5, cortesía de www.sparkfun.com 45

Ilustración 3.7 Pantalla uOLED 160G1, cortesía de www.4dsystems.com.au 46

Ilustración 3.8 Imagen en pantalla OLED, fuente www.4dsystems.com.au 47

Ilustración 3.9 uSD pantalla OLED, fuente www.4dsystems.com.au 50

Ilustración 3.10 Tablero de instrumentos 52

Ilustración 3.11 Sistema separado y ensamblado 55

Ilustración 3.12 Modelo funcional (CAD) 55

Ilustración 4.1 Bluetooth dongle 58

Ilustración 4.2 Prueba de laboratorio (PC) 59

Ilustración 4.3 Prueba de campo (automóvil) 61



Índice de diagramas

Diagrama 1.1 Clasificación de Pantallas.....	13
Diagrama 2.1 Diagrama funcional.....	21
Diagrama 2.2 Algoritmo.....	26
Diagrama 3.2 Conexión del bluetooth.....	35
Diagrama 3.3 Conexión no-handshake.....	35
Diagrama 3.4 Interfaz vehículo - procesamiento.....	36
Diagrama 3.5 Algoritmo general.....	38
Diagrama 3.6 1º Parte del algoritmo.....	39
Diagrama 3.7 2º Parte del algoritmo.....	40
Diagrama 3.8 3º Parte del algoritmo.....	42
Diagrama 3.9 4º Parte del algoritmo.....	43
Diagrama 3.10 Conexión pantalla - ATmega2650 - Antena Bluetooth.....	47
Diagrama 3.11 Distribución de la información en la pantalla OLED.....	51
Diagrama 3.12 Conexión del modelo funcional.....	54
Diagrama 6.1.1 Encapsulado ELM327.....	71

Índice de tablas

Tabla 1.1 Comparativa de sistemas HUD.....	12
Tabla 2.1 Jerarquización de requerimientos.....	21
Tabla 2.2 Matriz de conceptos.....	22
Tabla 2.3 Escala de evaluación.....	23
Tabla 2.4 Matriz de selección.....	24
Tabla 3.1 Parámetros de identificación.....	31
Tabla 3.2 Parámetros de Bluetooth.....	34
Tabla 3.3 Parámetros de comunicación (escáner - módulo bluetooth).....	35
Tabla 3.4 Características de la pantalla OLED.....	44
Tabla 3.5 Conector CN3.....	46
Tabla 3.6 Descripción de un comando para visualizar un texto en pantalla OLED.....	48
Tabla 3.7 Comparativa de configuraciones.....	53
Tabla 4.1 Automóviles para pruebas.....	60
Tabla 4.2 Comparativa entre automóviles.....	62

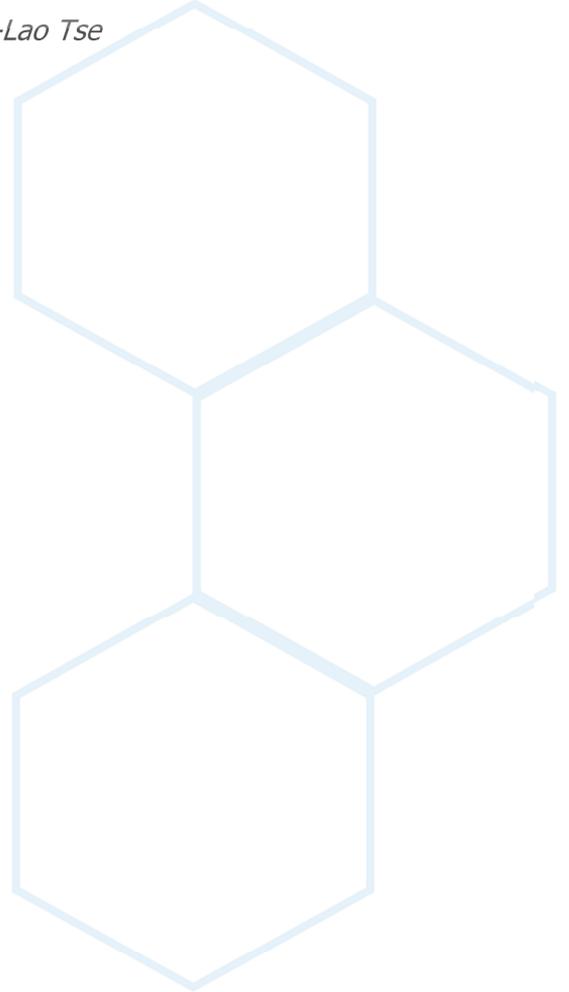
Índice de ecuaciones

Ecuación 1. PID's de RPM.....	32
Ecuación 2 Tiempo de descarga.....	54

INTRODUCCIÓN

"Un viaje de mil millas comienza con un paso"

-Lao Tse





Introducción

El hombre por naturaleza siempre ha buscado mejorar el entorno en el que vive, tratando de buscar la manera de satisfacer sus necesidades y deseos. La tecnología es una de las principales herramientas que ha utilizado la humanidad para cumplir gran parte de sus necesidades y deseos, por esa razón es uno de los sectores más competitivos, además de ser un gran generador de empleos y utilidades.

La necesidad del hombre de transportarse ha generado empresas transnacionales dedicadas exclusivamente al transporte. Dentro de dicha industria encontramos a la industria automotriz que se dedica al diseño, desarrollo, fabricación y venta de automóviles, esta industria compite día a día por el contar con la tecnología más nueva e innovadora. Existen países dedicados a la producción de vehículos, principalmente asiáticos, norteamericanos y europeos. México es un país productor de automóviles generando 2.680.037 unidades en 2011 encontrándose dentro de los 20 principales países productores de vehículos con motor [1].

Aunque México es uno de los principales productores, se enfoca principalmente al ensamble de vehículos, el diseño y desarrollo es menor comparado con otros países esto se debe a que la mayoría de las empresas que laboran en territorio mexicano son extranjeras, aun así en México demuestra su capacidad de diseño y desarrolló en el automóvil Mastretta [2]. Considerando lo anterior se piensa que es de suma importancia profundizar aun más en el campo de diseño y desarrollo de productos dentro del área automotriz, por esa razón el presente trabajo que lleva por título “PANTALLA OLED PARA INFORMACIÓN DEL TABLERO DE INSTRUMENTOS EN VEHÍCULOS” el cual busca demostrar la incursión en este campo con un proyecto que se encargará de visualizar la información del vehículo, como velocidad, revoluciones del motor, etc., en un pantalla de tecnología *OLED*, colocándola en el parabrisas.

El presente trabajo está compuesto por 5 capítulos y anexos donde se documentará el desarrollo de proyecto:

En el capítulo uno se abordaran los antecedentes necesarios para la comprensión de la tesis hablando de la industria automotriz, los módulos de seguridad y confort implementados en los automóviles, el historia y desarrollo de los sistemas *Head-Up Display* además de la tecnología *OLED* y algunas de sus aplicaciones.

El capítulo dos se enfoca en el diseño conceptual que se realizó para el sistema de tablero de instrumentos, el esta parte se fija el objetivo del proyecto además de definir los requerimientos del sistema, posteriormente se generan posibles soluciones además del criterio de selección para las mismas, concluyendo con una arquitectura del sistema.



El capítulo tres se desglosa a detalle el desarrollo de un modelo funcional del sistema de tablero de instrumentos, en él se especifica cómo se encuentra compuesto el sistema con sus respectivas funciones además de los elementos utilizados, la manera en que se trabajó con ellos y cómo interactúan entre sí mismos.

El capítulo cuatro condensa la información recabada de los resultados de las pruebas a las cuales fue sometido el sistema durante el desarrollo del proyecto, en éste se especifica la clase de pruebas, las simulaciones y además de los vehículos donde el sistema fue probado.

El capítulo 5 se encuentran las conclusiones que se generaron al finalizar todo el proyecto además de una sección dedicada al posible trabajo que se podría realizar en un futuro al continuar con este proyecto.

Posteriormente se encontrará con una serie de anexos que complementan la documentación del presente trabajo, como serian información complementaria, diagramas, tablas y planos.

La finalidad de este trabajo es demostrar la aplicación de los conocimientos adquiridos durante la carrera de ingeniería además de comprobar que en México se pueden incursionar en el desarrollo y diseño de módulos de confort y seguridad para la industria automotriz aplicando nuevas tecnologías como son las pantalla de tipo *OLED*.

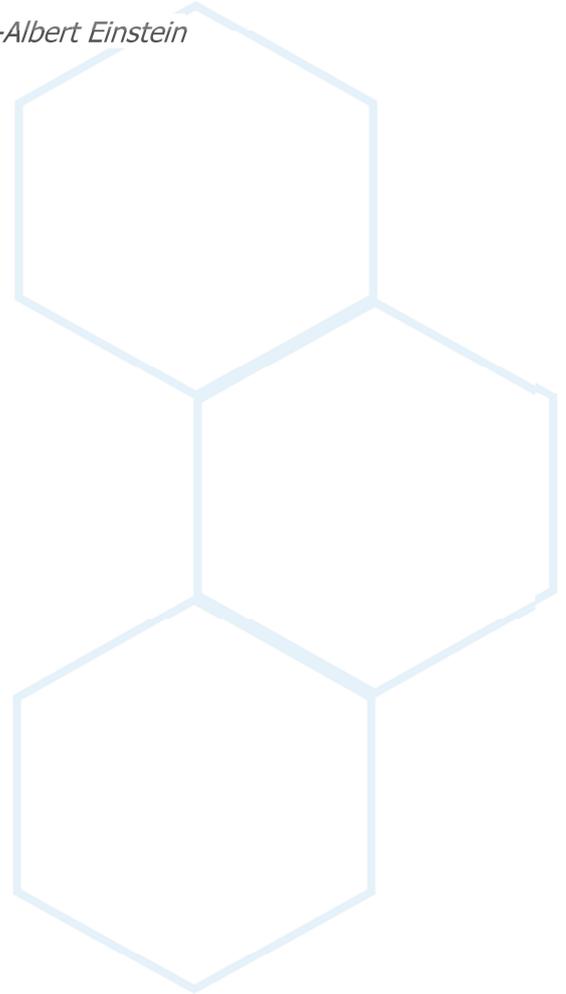


ANTECEDENTES

Capítulo 1

*"Si buscas resultados distintos, no
hagas siempre lo mismo"*

-Albert Einstein



- 1.1. - *Industria Automovilística.*
- 1.2. - *El automóvil del Futuro.*
- 1.3. - *Sistema Head-Up Display (HUD).*
- 1.4. - *Pantalla OLED (Organic Light-Emitting Diode).*
- 1.5. - *Planteamiento del Problema.*





1.1 La industria automovilística

La industria automovilística se especializa en el diseño, desarrollo, fabricación, ensamblaje, comercialización y venta de automóviles. Es una gran generadora de empleo generando mano de obra directa y toda una industria paralela de componentes. En 2011 se estima que se produjeron alrededor de 80 millones de unidades de vehículos de motor alrededor del todo el mundo según las estadísticas de *Organisation Internationale des Constructeurs d'Automobiles* (OICA) [1].

Al ser una industria de nivel mundial se encuentra conformada por grandes marcas como: *Toyota Motors Corporation, General Motors Corporation, Volkswagen Group, Ford Motor Company, Honda Motor Company, Chrysler Group, Nissan Motors, etc.* Reconocidas marcas de prestigio que buscan el auto ideal y compiten día a día para conseguirlo.



Ilustración 1.1 Logotipos industria automotriz, fuente www.autox4.com

Hoy en día esta industria avanza a pasos agigantados con respecto a la tecnología implementada en sus automóviles, buscando satisfacer las necesidades de los usuarios. La industria automotriz procura varios aspectos como la eficiencia del automóvil, la comodidad del mismo, el entretenimiento, la comunicación y una gran gama de parámetros para su diseño. En el transcurso de los años se han implementados nuevas tecnologías a los vehículos para mayor seguridad y confort, un claro ejemplo del pasado, son las bolsas de aire (*Airbag*).

La *Airbag* cual fue patentada en Octubre de 1971 por la marca de *Mercedes-Benz*, pero no fue hasta 10 años después donde se logró ver implementado en el modelo *Mercedes-Benz Clase S W126* [3]. El desarrollo de estos módulos han revolucionado la manera de percibir el automóvil de una simple máquina con finalidad de transporte a un elemento de confort y placer.



El automóvil en la actualidad cuenta con amplia gama de módulos de confort y seguridad. Esta industria busca la vanguardia en nuevas tecnologías para implementarlo en sus vehículos, ¿Cuál es la percepción de la industria automotriz con el futuro del automóvil?

1.2 El automóvil del futuro

Amory Lovins científico y director del *Rocky Mountain Institute*, experto en el campo de la energía, plantea su visión del automóvil del futuro: citando

“El auto del futuro debe ser menos contaminante, más seguro, más fuerte, más durable, más liviano, más cómodo y con mucho mayor rendimiento.” [4]

Importantes marcas de la industria automovilística coinciden en una tendencia sobre cómo será el vehículo del futuro, una de las principales características que se ve reflejada en los diseños de los próximos modelos y que hoy en día se están empezando a implementar son las alternativas para el uso del combustible. Buscar otras maneras de proporcionar energía a los automóviles que no dependa de los hidrocarburos, es una necesidad que la industria automovilística busca resolver lo antes posible, unas ideas de la industria para sustituir el combustible fósil son el uso del hidrógeno como combustible, o la tendencia que al parecer la mayoría de las marcas creen que es lo más viable, el auto eléctrico.

1.2.1 Módulos de innovación de confort y seguridad

Los automóviles se han convertido en instrumentos inteligentes y autónomos con el transcurso de los años, para conseguir esto se han implementan módulos que innoven los vehículos. Unos ejemplos de estas tecnologías y que posiblemente se pretendan incluir en los autos que saldrán a la venta en un futuro son:

Control de crucero: Sistema que permite mantener la velocidad constante y soltar el pie del acelerador. Muy útil para largos trayectos por autopista, se anula automáticamente por razones de seguridad en cuanto se toca el pedal del freno.

Cambio de seis marchas: Este sistema permite escalonar mejor las marchas, aprovechando mejor la potencia del motor, reducir el consumo y aumentar el confort del recorrido. Normalmente sólo disponible en modelos de alta gama o de carácter deportivo.



Sensor de lluvia: Es un dispositivo situado en el parabrisas que detecta la presencia de agua y pone en marcha las escobillas al ritmo adecuado según la intensidad de la lluvia.

Sensor de luces: En un sistema que cuenta con una célula que detecta la intensidad de la luz en el exterior del vehículo y conecta automáticamente las luces cuando oscurece.

Sistemas Hella de Iluminación en Curva: El sistema *DynaView de Hella* consta de unos faros adicionales que dan un haz de luz oblicuo, para iluminar mejor la carretera si hay curvas o en cruces.

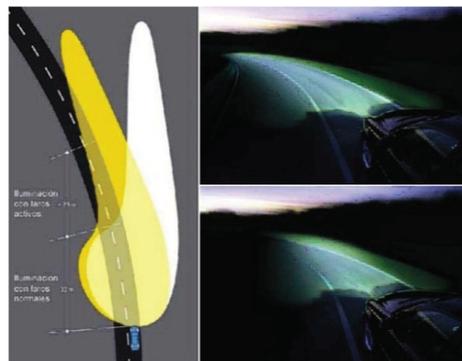


Ilustración 1.2 Sistema Hella de Iluminación en curva, fuente www.hella-shop.com.mx

Volante multifuncional: La automotriz *Volvo* además de otras desarrollaron un volante multifuncional dotado con botones para controlar el radio, teléfono móvil y otros accesorios. Importante para la seguridad evitando distracciones a la hora de buscar el volumen de la radio o responder una llamada.

Volvo desarrollo un sistema que detecta peatones y autos a bajas velocidades, este sistema se llama "*City Safety*", ayuda a evitar colisiones durante la conducción, este sistema está siendo muy aceptable por los consumidores.



Ilustración 1.3 Sistema City Safety, fuente www.volvocars.com/us



Toyota y Nissan al igual que otras empresas desarrollan un sistema *Anti-drunk Driving Technology* (ADT) detectando si el conductor está bajo los efectos del alcohol. El sistema ADT consiste en detectar por medio del sudor de las manos los grados de alcohol ingeridos por el conductor, esto con ayuda de un sensor ubicado en la palanca de las velocidades. El sistema ADT inhabilita el arranque del motor si el usuario supera los límites de alcohol permitidos.



Ilustración 1.4 Sistema ADT Nissan, fuente www.nissan-global.com

1.3 Sistema Head-Up Display (HUD)

1.3.1 Definición del sistema HUD

El *head-up display*, o HUD, es una pantalla transparente que presenta información al usuario de tal forma que éste no debe cambiar su punto de vista para ver dicha información. El origen del nombre proviene del hecho de que el usuario puede ver la información necesaria con la cabeza erguida (*head up*) mirando al frente, en vez de bajar la cabeza para revisar los instrumentos. Aunque su desarrollo inicial fue para las aeronaves con fines militares en la Segunda Guerra Mundial, el primer prototipo de aviones que incluyó este sistema fue el “B-103 *Buccaneer*” en Abril de 1958 [5] actualmente se utilizan estos sistemas en la aviación civil, automóviles, etc.



Ilustración 1.5 Sistema Head-Up Display, fuente aerotecnologia.blogspot.mx



Se busca adaptar esta tecnología a los automóviles y se han realizado varios prototipos. Los sistemas de información en el parabrisas son aún poco habituales en los autos enfocados más a la línea de lujo, algunos modelos ya equipan un sistema *HUD* básico, como se muestra en el *Prius* de Toyota.



Ilustración 1.6 HUD Toyota, fuente www.toyota.com.au/prius

Diferentes fabricantes están apostando por este sistema y es muy probable que en unos diez años esté mucho más comercializado. La ventaja del sistema es clara: permite visualizar datos fundamentales durante la conducción frente a la línea de los ojos, sin requerir que se aparte la vista de la carretera ni medio segundo.

1.3.2 Principio de funcionamiento

HUD se basa en el principio de reflexión de la luz sobre un vidrio. Así que es básicamente un sistema de proyección. Lógicamente la reflexión tal cual no suele resultar en imágenes siempre nítidas o contrastadas, así que suelen añadirse algunos factores más a la ecuación. Uno de ellos es aumentar la reflectividad del soporte, por ejemplo haciendo que este sea algo más opaco, oscuro o vidrios polarizados. En otras ocasiones se proyecta sobre una lámina específica adherida al vidrio.

1.3.3 Tipos de sistemas HUD

En el ámbito militar se desarrollaron dos tipos de *HUD*:

- Los **fijos** la información aparece en el parabrisas del automóvil.
- Los **de casco**, llamados *HMD*, por *Helmet Mounted Display* que muestran la información en la propia visera del casco o bien en una lente frente a uno de los ojos.



1.3.4 Partes que componen el sistema

Un sistema *HUD* consta por tanto de tres partes fundamentales:

- El combinador que es la superficie sobre la que se proyecta la imagen.
- El proyector el que genera la imagen y la dirige hacia el combinador.
- La unidad de computación que procesa los datos de las diferentes fuentes y las organiza para su proyección.

1.3.5 Generación de la imagen

La imagen, dígitos o símbolos que se proyectarán sobre el parabrisas, se pueden generar de múltiples maneras:

- Generados por una pequeña pantalla CRT (tubo de rayos catódicos) que se reflejaba en un espejo y de ahí al parabrisas.
- Un láser de barrido puede generar las imágenes directamente sobre una superficie transparente.
- Últimamente también se pueden utilizar pantallas LCD.

1.3.6 HUD como equipamiento de fábrica

Algunos modelos de automóviles ya equipan, un sistema de información en el parabrisas. Lo normal es que los datos se proyecten delante del conductor a una altura superior a la del cuadro de instrumentos. Oculto desde la visual del conductor, hay un hueco en la visera que cubre el cuadro de instrumentos en el que se sitúa el proyector que generará la imagen sobre el parabrisas. En el parabrisas puede notarse, según la luz, y si se tiene buen ojo, un pequeño recuadro con un matiz o tono ligeramente. La proyección se suele realizar en monocromo pero ya empiezan a aparecer sistemas que combinan tres o cuatro colores. Lo mínimo es visualizar la velocidad, pero también se suele mostrar, testigos de luces o cambio, e incluso indicaciones del navegador GPS.



Ilustración 1.7 HUD BMW, fuente www.bmw.com



Se considera que fue *General Motors*, en 1988, el primer fabricante en implementar un sistema *HUD* en un automóvil, e incorporar uno a color en 1998 con el *Corvette C5*. *Toyota*, *Audi*, *BMW*, *Lexus* y *Honda* también tienen varios modelos, así como *Citroën* con el *C6*. El fabricante francés *Peugeot* ha optado por un sistema *HUD* ligeramente diferente al de estas otras marcas, el *Peugeot 3008* en lugar de proyectar directamente sobre el parabrisas, lo hace sobre un pequeño panel de policarbonato translúcido detrás del cuadro de instrumentos, que se puede ocultar si se desea [6].



Ilustración 1.8 HUD Peugeot, fuente www.peugeotforums.com

1.3.7 Propuestas de futuro en HUD

La tendencia de los últimos años indica que en un futuro el *HUD* será más multimedia, integrarán múltiples funciones y dispositivos digitales como el navegador *GPS*, Internet, teléfono, etc. *General Motors* pretende implementar estas funciones con realidad aumentada, ya sea para aumentar la información, como por ejemplo la presentada en un *GPS*, indicando de manera gráfica la ruta seleccionada previamente, la ubicación de gasolineras, restaurantes, etc. *GM*® considera que esta tecnología estará implementada para los modelos 2020 [7].

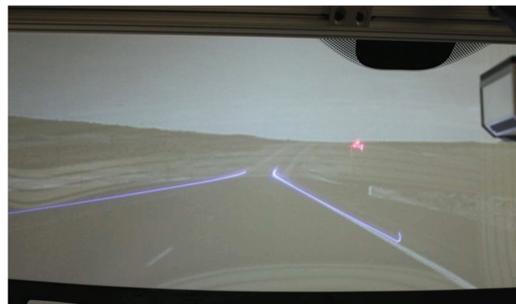


Ilustración 1.9 Realidad Aumentada por General Motors, fuente www.autopasion.com

Audi también está investigando acerca del sistema *HUD*, con su sistema *Aida 2.0* propone una proyección tridimensional y holográfica, de los mapas y rutas del navegador, además de otros datos relativos a la realidad aumentada, como eventos, ofertas, menús, etc [8].

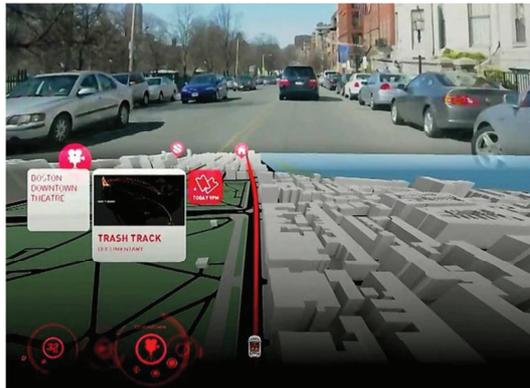


Ilustración 1.10 Sistema Aida 2.0, fuente www.motorpasionfuturo.com

Tabla 1.1 Comparativa de sistemas *HUD*

Sistema <i>HUD</i>	Ventajas	Desventajas
Toyota Pruis HUD	Utiliza la reflexión de una pequeña pantalla sobre el parabrisas. Ya se encuentra implementado en algunos autos comerciales.	Muestra poca información (Velocidad). La información se visualiza de tamaño pequeño. La información se visualiza con bajo contraste.
GM HUD	Utiliza la proyección de una cámara para visualizar la información. Muestra más información en el parabrisas. Implementa la tecnología de realidad aumentada.	Por los componentes utilizados el precio se eleva. Contiene un mayor número de elementos en el sistema.
Audi Aida 2.0	Implementa la tecnología de realidad aumentada. Muestra mayor información sobre el parabrisas. Se muestran imágenes tridimensionales.	Se encuentra aún en etapa de prototipo. El precio al producirlo es mayor.
Peugeot HUD	Muestra mayor información al usuario La pantalla es ajustable en posición. La información mostrada puede ser ajustada para tener un mayor contraste. Se encuentra implementado en algunos autos.	Cuenta con una pantalla adicional para mostrar la información. Contiene un mayor número de elementos en el sistema.



1.4 Pantalla OLED (*Organic Light-Emitting Diode*)

1.4.1 Introducción

Organic Light Emitting Diode es el significado de la abreviatura *OLED*, la misma que podría traducirse como “diodo orgánico emisor de luz”. Los *OLEDs* son diodos de película delgada cuya capa emisora de luz está formada por un polímero orgánico, es decir, grandes moléculas compuestas de unidades químicas en cadena que son capaces de convertir energía eléctrica en energía luminosa cuando se sitúan entre dos electrodos. Por lo tanto la química tiene mucho que ver con tecnología. Los *OLEDs* se encuentran dentro de la clasificación de las pantallas emisoras de luz como se muestra en la siguiente ilustración.

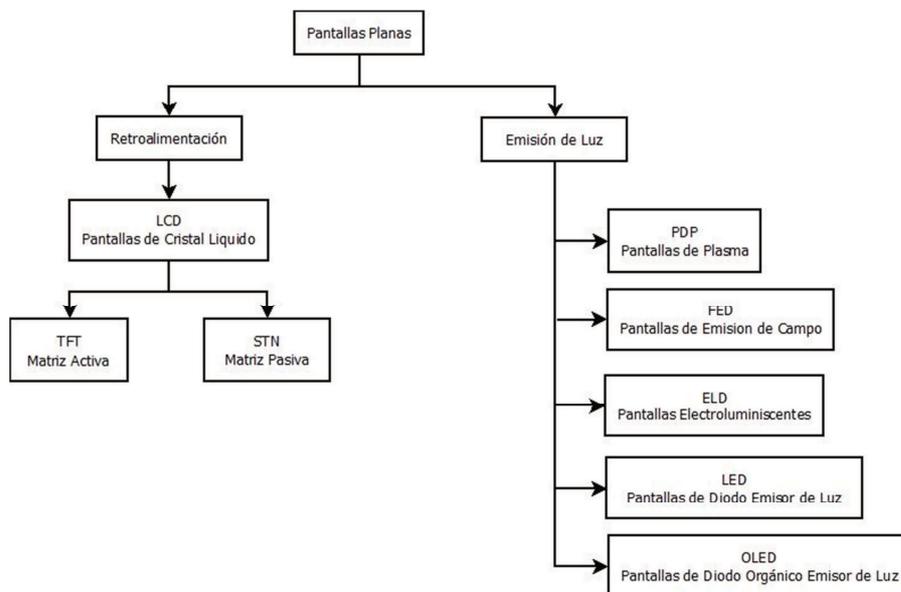


Diagrama 1.1 Clasificación de Pantallas

El hecho de que sean orgánicos demanda un notable avance. Hasta ahora, los *LEDs* se hacían con semiconductores inorgánicos como es el caso del silicio, arseniuro de galio, arseniuro de indio, etc. Lo que implicaba una serie de limitaciones en cuanto a rendimiento. La complejidad de las moléculas semiconductoras orgánicas permite conseguir rendimientos notablemente superiores de forma mucho más sencilla, una vez que se sabe cómo fabricarlos.



Las pantallas *OLED* se basan en un principio foto-químico por el cual ciertas moléculas orgánicas al ser excitadas por una corriente eléctrica emiten luz, este fenómeno tiene cierta similitud con el conocido fenómeno piezo-eléctrico al aplicar una corriente eléctrica se produce una deformación del material que lo hace vibrar a su frecuencia de resonancia. Estudios han demostrado que el espesor de un *OLED* varía entre 100 y 500 nanómetros, es decir unas 200 veces más delgados que el cabello humano.

Las pantallas *OLED* son más delgadas y su peso es menor ya que no requieren iluminación de contraluz. También proveen un amplio ángulo visual de aproximadamente 170°, inclusive en luz brillante, y requieren sólo entre dos y diez voltios para operar. Las nuevas tecnologías construidas sobre *OLED* incluyen las que se están utilizando para el desarrollo del papel electrónico es decir pantallas que se puedan doblar, enrollar y transportar fácilmente [9].

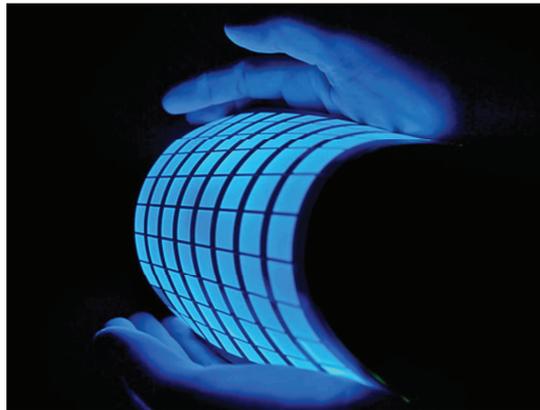


Ilustración 1.11 *Organic Light Emitting Diode (OLED)*, fuente www.bombillasoled.com

El único problema actual que presentan las pantallas *OLED* es que la duración de vida, se estima en 20.000 horas cuando los píxeles azules se oscurecen antes que los otros, En el momento en que los colores mueren, la pantalla adquiere un tono magenta, de manera que por ahora, este tipo de pantallas aparecerán en productos con ciclos de vida cortos. Pero ya se están desarrollando pantallas de hasta 150.000 horas de vida.

Se estima que a mediano plazo, las pantallas *OLED* ingresarán en el mercado desplazando al pantallas de cristal líquido (*LCD*) y revolucionando la calidad visual de todos los dispositivos portátiles. Desde el descubrimiento en 1987 de los diodos orgánicos emisores de luz el interés en este trabajo ha aumentado rápidamente. Las ventajas de los *OLEDs* son imposibles de pasar por alto, siendo los *OLEDs* son extremadamente delgados, muy livianos y ahorradores de energía.



1.4.2 Historia

Los OLED se descubrieron en los años 1987 y 1990, por investigadores *Dr. Tang* en *Kodak* y el Prof. *Friend* en la Universidad de *Cambridge*. En 1979 *Chin Tang* descubrió la electroluminiscencia en el departamento de investigación de *Kodak*. Durante su trabajo con celdas solares él observó un resplandor azul en el material orgánico. En 1987 *Tang* y *Van Slyke* introdujeron el primer diodo emisor de luz de capas orgánicas delgadas. Se descubrió en 1990 la electroluminiscencia en los polímeros [9].

Las primeras pantallas se fabricaban colocando diodos verdes, rojos y naranjas en formas rectangulares que se hacían funcionar siguiendo unos patrones específicos, y se empleaban en letreros y marcadores electrónicos para comunicar mensajes sencillos.

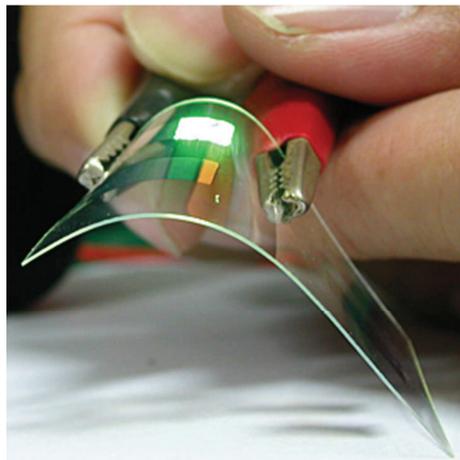


Ilustración 1.12 Primeros OLED, fuente comunicantropo.com/

Al sustituir el semiconductor inorgánico por un polímero orgánico de ahí la “O” se trata de un método más económico para producir diodos más versátiles y brillantes. Las nuevas técnicas de fabricación les permitieron a los científicos producir decenas de miles de píxeles a la vez, solucionando en gran medida el problema de la baja resolución de las anteriores pantallas. Lo que se hace es colocar uno o más polímeros especiales de tan sólo un micrón de grosor siguiendo determinados patrones sobre el sustrato elegido, para obtener pantallas de vidrio fino y muy ligero.

1.4.3 Componentes de los OLEDs

Los *OLEDs* son dispositivos semiconductores, compuestos por películas finas de moléculas orgánicas que crean luz por medio de la aplicación de electricidad.

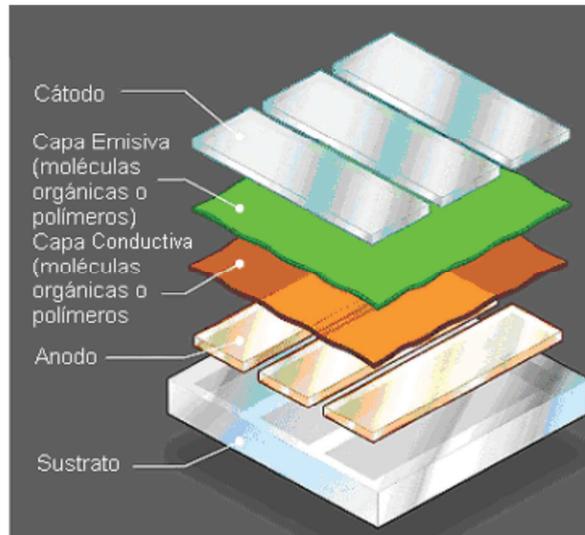


Ilustración 1.13 Estructura de una pantalla *OLED*, fuente www.hormigasolar.com

1.4.4 Descripción de los componentes de un *OLED*.

- El sustrato que puede ser de material como el plástico, cristal, hoja de aluminio y sirve como apoyo del *OLED*.
- El ánodo de tipo transparente remueve electrones agrega los electrones a los “agujeros” cuando una corriente atraviesa el dispositivo.
- Capas orgánicas: estas capas se hacen de moléculas o de polímeros orgánicos.
- Capa conductiva: esta capa está hecha de moléculas plásticas orgánicas que transportan “agujeros” desde el ánodo. El polyanilina es un polímero conductor semiflexible utilizado en las pantallas *OLED*.
- Capa emisiva: esta capa está hecha de las moléculas plásticas orgánicas diferentes a las utilizadas en la capa conductiva la cual transportan electrones desde el cátodo; aquí es donde se produce la luz. El polyfluorene es un material polímero que aunque no tiene un origen natural la química moderna permitió sintetizarlos y así poder utilizarlos en distintas aplicaciones de electrónica.
- El cátodo inyecta electrones cuando una corriente atraviesa el dispositivo, la transparencia de este depende del tipo de *OLED* [9].



1.4.5 Aplicaciones de las pantallas OLEDs

Las aplicaciones donde se introducen los *OLEDs*, inicialmente se utilizaron para mostrar mensajes monocromáticos, pero hoy en día se utilizan en distintos dispositivos electrónicos como televisores, computadoras, celulares, cámaras digitales, Reproductores de música, carátulas de autoestéreos, etc. Samsung, empresa coreana fundada en 1969, que se enfoca en la gama de artículos eléctricos-electrónicos considera que el futuro de las pantallas y demás dispositivos se encuentra en los *displays OLEDs*, presentando en 2008 los primeros prototipos de dispositivos que cuentan con este tipo de pantalla.



Ilustración 1.14 Laptop Samsung 14" OLED, cortesía de www.gadgetreview.com

Empresas de gran renombre han invertido millones de dólares en la investigación de este tipo de pantallas, para que en un futuro puedan ser implementadas en distintas aplicaciones. Hoy en día esta tecnología se utiliza exclusivamente para dispositivos electrónicos, pero en un futuro podremos verlas incluidas no sólo en esta industria, sino de igual forma en la publicitaria, industria textil, industria automotriz, etc. En 2012 se pretende sacar al mercado los primeros televisores que cuentan con esta tecnología, *LG* y *Samsung* fueron los primeros en anunciarlo [10].



Ilustración 1.15 TV OLED LG, fuente www.xataka.com



1.5 Planteamiento del problema

Hoy en día la tecnología avanza a pasos agigantados y la industria automotriz es una de las más innovadoras. Satisfacer las necesidades y deseos de los usuarios en los autos es el objetivo de esta industria, para ello se crean módulos de confort y seguridad. Un claro ejemplo de esto es que ya algunos modelos de automóviles cuentan con módulos de seguridad como bolsas de aire, detección de transeúntes, también módulos de confort como *Wi-fi*, sensores de luz, etc.

En el futuro se busca que los automóviles cuenten con la más alta tecnología, es necesario diseñar e implementar módulos innovadores. Se busca que sean más amigables con el medio ambiente, más inteligentes, autónomos posible y de bajo costo. Se pretende disminuir el espacio en el tablero o incluso cambiar su manera de mostrarse, como puede mostrarse en las tendencias de *Mitsubishi Motors®*, con el modelo del automóvil *Concept PX-MiEV*, resaltando principalmente su línea de autos eléctricos, teniendo un tablero iluminado, e equipado con un instrumental de fácil visualización [11].

¿Cómo mostrar al operario la información del automóvil de una manera distinta? Idealizar, diseñar o rediseñar algún módulo de confort para mostrar la información del tablero de un vehículo al usuario.

1.5.1 Necesidad

Se requiere el diseño de un sistema que visualice la información del tablero del automóvil sobre el parabrisas del mismo, sin perjudicar la visibilidad del operario.

1.5.2 Justificación

El desarrollo e innovación de nuevos productos es una parte fundamental para el avance en la tecnología, es importante que el hombre este incursionado en nuevas ideas. La industria automotriz es un claro ejemplo en diseño e innovación.

El diseño de un sistema que visualice la información del tablero de un vehículo en el parabrisas con apoyo de una pantalla OLED, conjunta la aplicación de varios conocimientos, como programación, selección de materiales, metodología de diseño, etc. En futuro se pretende que el módulo pueda ser aplicado para otras aplicaciones como visualizar el retrovisor, información proporcionada por un GPS, información multimedia, etc. Considerando que no sea algún elemento que perjudique la visibilidad el conductor y no comprometer su vida. En México se realiza muy poca investigación y desarrollo de nuevos productos comparados con otros países, por esa razón es muy importante incursionar en estos campos a una mayor proporción y de manera constante.



DISEÑO CONCEPTUAL

Capítulo 2

"La valentía no es la ausencia de miedo, sino el triunfo sobre él"

-Nelson Mandela



- 2.1. – Objetivo.
- 2.2. - Requerimientos.
- 2.3. -Funciones.
- 2.4. - Matriz de conceptos.
- 2.5. - Arquitectura de sistema.





2.1 Objetivo

“Realizar el diseño de un sistema que visualice la información del tablero de un automóvil en el parabrisas buscando que sea de bajo costo, adaptable a diferentes automóviles y no interfiera con la visibilidad de operario, llegando a un modelo funcional que demuestre su operación”.

2.1.1 Hipótesis

“Es posible diseñar un sistema no invasivo capaz de visualizar la información del tablero de un automóvil como la velocidad, las revoluciones por minuto, el nivel de combustible o alguna otra información, sobre el parabrisas del vehículo sin afectar la visibilidad del conductor”.

2.1.2 Alcances

Al final del proyecto se contará con un diseño del sistema “tablero de instrumentos para vehículo”, conteniendo con planos y diagramas de configuración además de un modelo funcional que ejemplifique aquellas características mínimas que validen la operación del sistema. Obtener un módulo que incluya una pantalla de tipo *OLED* como elemento del sistema, obteniendo una aplicación con este tipo de *displays* para posteriormente ser implementado en vehículos considerando como un factor esencial la visibilidad del operario.

2.2 Requerimientos y Necesidades

- El sistema visualizará la información del tablero en el parabrisas.
- El sistema será lo menos invasivo al automóvil.
- Se buscará que sea de bajo costo.
- El sistema permitirá al usuario contar con una buena visibilidad del camino.
- El sistema se mantendrá funcionando mientras el automóvil se encuentre encendido.
- La información se visualizará de una manera clara, legible y entendible.



Tabla 2.1 Jerarquización de requerimientos

Requerimiento	Importancia
Visualizar información	5
No invasivo al automóvil	4
Bajo costo	3
Mantener la visibilidad del camino	5
Tiempo de funcionamiento	4
Información clara, legible y entendible	5

2.3 Funciones del sistema

A continuación se mostrará un diagrama donde represente la interacción de las funciones entre ellas dentro del sistema de tablero de instrumentos:

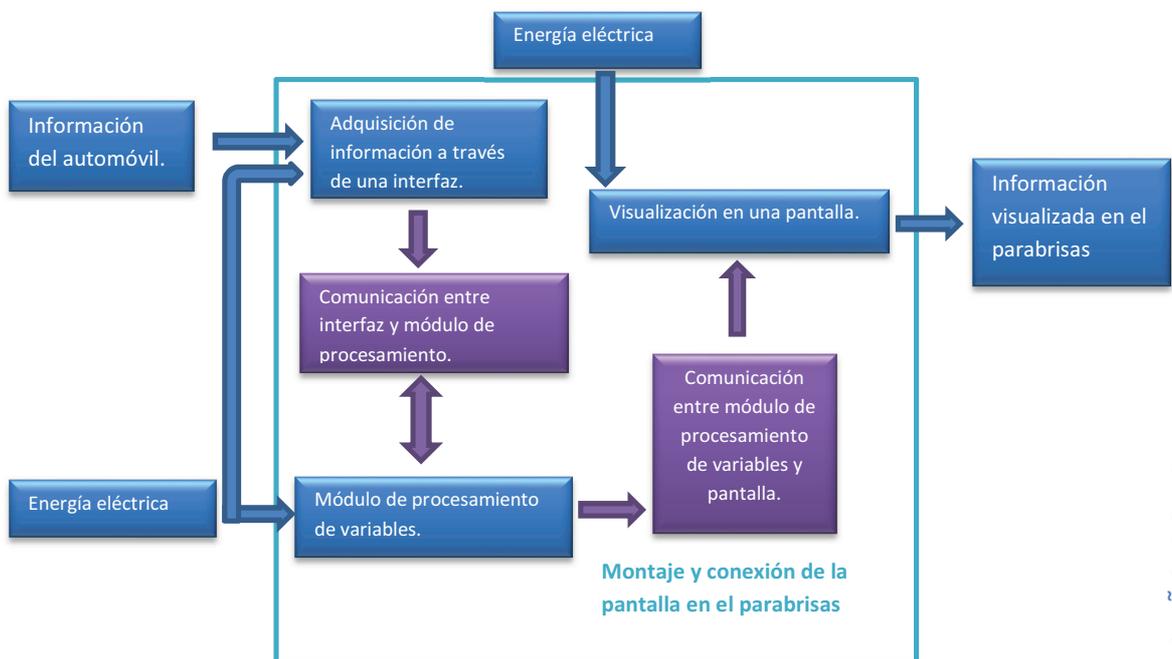


Diagrama 2.1 Diagrama funcional



2.3.1 Descripción de las funciones

Adquisición.- Esta función se encarga de buscar y obtener la información proveniente del automóvil.

Interfaces de comunicación.- La tarea será el comunicar elementos y acondicionar las señales hasta su visualización.

Procesamiento de variables.- Aquí se realizará el procesamiento necesario para la relación entre las señales obtenidas y la información que si visualizará en el parabrisas.

Visualización.- Se proyectará la información resultante en una pantalla ajustable al parabrisas.

Montaje.- Se encarga de elegir la más adecuada posición del sistema en el automóvil.

2.4 Matriz de conceptos

Se utilizará una metodología de diseño similar a la propuesta de Hans Gugelot alrededor de 1960. En este caso la selección del concepto a desarrollar para el sistema de tablero de instrumentos, se implementará la matriz de conceptos y selección, en base a los requerimientos establecidos en un principio del diseño. Una matriz de conceptos condensa todas las posibles soluciones para cada función del sistema, se realizaran una serie de combinaciones para obtener unas soluciones.

Tabla 2.2 Matriz de conceptos

Adquisición	Interfaces de comunicación	Procesamiento de información	Visualización	Montaje
Sensores ★	Bluetooth ★	Amplificadores	Proyección	Sujeción ★
Tablero ★	Serial ★	FPGA ★	Pantalla LCD	Adhesivo ★
Computadora (ECU) ★	Wi-Fi	Microcontrolador ★	Pantalla OLED ★	
		Software(PC)	Reflexión ★	
		DAQ ★		



★ *Concepto 1:* Adquisición de información a través de los sensores del automóvil, con interfaces de comunicación de tipo alámbrica serial, DAQ como dispositivo de procesamiento de información, el elemento para visualizar Pantalla OLED con un elemento de montaje adhesivo.

★ *Concepto 2:* Adquisición de información a través del tablero de instrumentos del automóvil, con interfaces de comunicación de tipo alámbrica serial, FPGA como dispositivo de procesamiento de información, el elemento para visualizar Pantalla OLED con un elemento de montaje sujeción.

★ *Concepto 3:* Adquisición de información a través de la computadora del automóvil (ECU) con interfaces de comunicación de tipo inalámbrica bluetooth, microcontrolador como dispositivo de procesamiento de información, el elemento para visualizar Pantalla OLED con un elemento de montaje adhesivo.

2.4.1 Matriz de selección

Para la matriz de selección se implementará una escala de evaluación:

Tabla 2.3 Escala de evaluación

Desempeño	Calificación
Insuficiente	1
Deficiente	2
Buena	3
Muy Buena	4
Excelente	5

La matriz de selección evalúa las distintas soluciones obtenidas en la matriz de conceptos, se utilizó como criterios de selección los requerimientos de diseño otorgándoles a cada uno un peso en porcentaje, para posteriormente calificar a cada solución con la escala anterior.



Tabla 2.4 Matriz de selección

		Concepto 1 ★		Concepto 2 ★		Concepto 3 ★	
Crterios de selección	Peso (%)	Calif	Puntuación Ponderada	Calif	Puntuación Ponderada	Calif	Puntuación Ponderada
<i>Visualizar información</i>	25 %	4	1	4	1	4	1
<i>No invasivo al automóvil</i>	15 %	1	0.15	2	0.30	3	0.45
<i>Bajo costo</i>	10 %	2	0.20	1	0.10	3	0.30
<i>Mantener la Visibilidad del camino</i>	25 %	3	0.75	3	0.75	3	0.75
<i>Tiempo de operación con el vehículo encendido</i>	10 %	3	0.30	4	0.40	3	0.30
<i>Información clara, legible y entendible</i>	15 %	3	0.45	3	0.45	3	0.45
Puntuación Final	100 %		2.85		3.00		3.25
Selección de solución		NO		NO		SI	

Se concluye que el concepto N° 3 es la que se desarrollará para el sistema de tablero de instrumentos, obteniendo una mejor calificación al ser evaluada con requerimientos.

2.5 Arquitectura de sistema

Una vez seleccionada la solución que se desarrollará para el sistema lo siguiente será obtener la arquitectura del sistema donde se explicaran las etapas que lo compondrán.

- A. Etapa de adquisición de información.
- B. Interfaz de comunicación vehículo – procesamiento.
- C. Módulo de procesamiento de información.
- D. Interfaz de comunicación procesamiento – pantalla.
- E. Visualización de información en pantalla.
- F. Montaje del sistema en el automóvil.



2.5.1 *Etapas de adquisición de la información:*

La adquisición se encarga de obtener la información del automóvil, como velocidad, rpm, etc. Como solución la adquisición será por medio de la *ECU (engine control unit)* o computadora del vehículo, se tiene un escáner que permite establecer la comunicación con el automóvil, generalmente estos se conectan mediante el conector *OBD-II* y utilizan algún protocolo de comunicación como por ejemplo *CAN, ISO 9141-2*, etc. Los parámetros que se deben de considerar para la selección del dispositivo son:

- Compatible con distintas marcas de automóviles.
- El protocolo de comunicación con la computadora del vehículo.
- Manejo de información del automóvil.
- Forma de alimentación eléctrica del escáner.
- Costo del escáner.

Una vez seleccionado el escáner se debe de estudiar su principio de funcionamiento para poder adquirir la información del automóvil, posteriormente transmitirla por la interfaz de comunicación hasta el módulo de procesamiento.

2.5.2 *Interfaz de comunicación vehículo - procesamiento:*

La interfaz se encargará de realizar la comunicación entre el escáner y el módulo de procesamiento de información. Dependiendo de las condiciones se pueden tener conexiones de tipo *bluetooth, serial, wi-fi*, etc. En específico para esta interfaz de comunicación se considera el tipo de escáner que se implementará. Se utilizará una conexión de tipo inalámbrica como se indica en el concepto N°3 del sistema, considerando un automóvil de tipo comercial de 4 plazas, la distancia que existe entre el conector *OBD-II* donde se conectará el escáner y el parabrisas no es mayor a 2 metros, una opción viable es la comunicación *bluetooth* teniendo como ventajas el costo y principio de funcionamiento.

2.5.3 *Etapas de procesamiento de información*

Esta etapa implementará un microcontrolador capaz de realizar el algoritmo necesario para procesar la información obtenida del vehículo, posteriormente dicha información se proyectará en la pantalla. Básicamente el algoritmo que efectuará la etapa de procesamiento de información se resume a:

1. Iniciar la comunicación con el automóvil
2. Recibir la información del automóvil.
3. Discriminar y seleccionar la información.
4. Procesar tipo de variables.
5. Transmitir la información depurada.
6. Seleccionar y transmitir los comandos para el control de la pantalla OLED.



A continuación se ejemplificará el algoritmo general para el sistema de tablero de instrumentos de vehículos con pantalla OLED, con un diagrama de flujo:

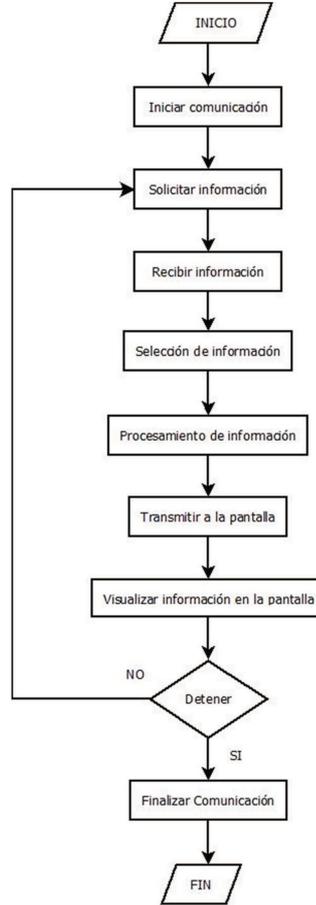


Diagrama 2.2 Algoritmo

La selección del dispositivo dependerá de ciertos parámetros, como son velocidad de transmisión, puertos de comunicación, lenguaje de programación, etc.

2.5.4 Interfaz de comunicación procesamiento – pantalla

Para la transmisión de la información del módulo de procesamiento a la pantalla se necesitará algún elemento que facilite el controlar la pantalla OLED, por lo general existen controladores gráficos que permiten una gama de conexiones de tipo Serial, VGA, DVI, etc. Se deberá elegir una pantalla que permita un controlador que pueda ser conectado con el módulo del procesamiento y así transmitir la información.



2.5.5 Visualización de la información en la pantalla

En esta etapa se debe de considerar la pantalla de tipo OLED, además de definir la manera y forma que se mostrará la información dentro de la pantalla para esta aplicación se necesita contar con características específicas para el sistema de tablero de instrumentos las cuales son las siguientes:

- Flexible.
- Transparente a 80%.
- Delgada.
- Color.

Samsung ha investigado e implementado mucho en este tipo de tecnología incluso a otorgado un nombre específico a pantallas con características similares, “*YOUM®*” es el nombre con el cual se les conocerá a estas pantallas. *Samsung* espera iniciar su producción en el segundo semestre del año 2012 [12].

2.5.6 Montaje del sistema en el Automóvil

Esta etapa busca elegir la posición más adecuada del sistema dentro del automóvil, para esto se deberá de conocer el tipo de conexiones con el que contará todo el sistema, además de los elementos que sujetarán a la pantalla *OLED*, al módulo de procesamiento, etc.

Los parámetros que se deben de considerar para ubicar la mejor posición del sistema son:

- La visibilidad del camino al conductor.
- El tiempo que tarde el conductor en observar la información.
- La alimentación eléctrica del sistema.
- El tipo de conexiones.
- La dificultad de montar el sistema en el automóvil.

Cabe mencionar que dependiendo del vehículo en el cual se instale el sistema puede existir variantes.

Capítulo 3

"Nunca ande por el camino trazado, porque él conduce adonde ya fueron otros"

-Alexander Graham Bell



- 3.1. - Preámbulo.*
- 3.2. - Adquisición de información.*
- 3.3. - Interfaz de comunicación vehículo - procesamiento.*
- 3.4. - Módulo de procesamiento de información.*
- 3.5. - Interfaz de comunicación procesamiento - pantalla*
- 3.6.- Visualización de la información en la pantalla.*
- 3.7.- Montaje del sistema en el automóvil.*





3.1 Preámbulo

Es importante demostrar la fiabilidad del proyecto, desarrollando un modelo funcional que ejemplifique aquellas características mínimas que validen la operación del sistema y su funcionabilidad. Tomando como guía la arquitectura del sistema desarrollada en el diseño. Una de las principales dificultades para la elaboración de un modelo funcional es el no contar con los elementos ideales para su fabricación, por esas circunstancias se deben de remplazar algunos elementos por otros considerado que cuenten con el mismo principio de funcionamiento.

Las pantallas *OLED* son un tecnología relativamente nueva aún se encuentran en etapa de desarrollo y son pocos los dispositivos que cuentan con alguna de ellas. Aun así la venta y aplicación de *displays OLED* con características específicas como la transparencia y flexibilidades son muy escasa aún. Dificultando en gran manera el desarrollo de este sistema, por desgracia no se contó con un panel de dichas características para el desarrollo del proyecto provocando que el modelo funcional se sustituya con una pantalla *OLED* a color pero sin las características de transparencia ni flexibilidad.

La selección de la pantalla provocaría la elección de algunos componentes, principalmente aquellos que se encuentren en contacto con la misma. Se espera que en una segunda versión de este sistema de tablero de instrumentos para vehículos ya pueda implementarse un *display* con las características establecidas desde un principio, obteniendo un resultado aún más favorable.

3.2 Adquisición de Información

Un requerimiento establecido para el sistema, fue que la adquisición de información debe ser lo menos invasiva al vehículo, y la solución que se obtuvo de la matriz de selección fue que dicha adquisición se deberá realizar a través de la computadora del automóvil (*ECU*) con apoyo de un escáner.

3.2.1 Elementos para mostrar en el parabrisas.

El tablero de instrumentos es un elemento del vehículo que contiene una gran gama de información, la cual es indispensable para el monitoreo del automóvil, como primera versión del sistema se han elegido los siguientes indicadores:



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO



- Velocímetro: Instrumento que indica la velocidad media de un vehículo.
- Tacómetro: Instrumento que indica la velocidad con la que gira el eje del motor del vehículo.
- Combustible: Instrumento que indica el nivel de combustible del automóvil.



Ilustración 3.1 Elementos del tablero, fuente www.themotorreport.com.au

Para obtener la información se requiere un escáner que cuente con las características de compatibilidad, bajo costo, el tipo de conectividad, etc. Después de realizar un estudio de distintos escáners que se podrían utilizar para el sistema se seleccionó uno el escáner lleva por nombre *ELM327 interface*, dicho escáner se conectar por la entrada *OBD-II* del automóvil para entablar la comunicación con la *ECU* del vehículo, para mayor información consultar *apéndice A*.

El *OBD (On Board Diagnostics)* es un sistema de diagnóstico a bordo en vehículos. Actualmente se emplean los estándares *OBD-II* que aportan un monitoreo y control completo del motor y otros dispositivos del vehículo. Los vehículos poseen una norma diferente, regulada por la *SAE*, conocida como *J1939*.



Ilustración 3.2 Escáner ELM327, fuente www.diagnosticoautomotor.com



3.2.2 Funcionamiento del escáner ELM327

El escáner cuenta con un encapsulado ELM327 de ahí proviene el nombre del escáner, que se encarga de establecer la comunicación con la computadora del automóvil a través de unos comandos AT, transmitiendo un código que genera una respuesta en forma de una cadena hexadecimal, la cual se procesa para obtener el valor real, a estos comandos con su información resultante se le conoce como PID's (Parameter ID's) para mayor información consultar el apéndice B.

Dicha respuesta se obtiene de sensores que se encargan de monitorear los parámetros del vehículo. Dependiendo del modelo de automóvil se puede o no monitorear algunos parámetros de identificación, esta sobre entendido que entre más reciente sea el modelo del carro tiene un mayor número de sensores y por lo consiguiente una mayor gama de información disponible.

A continuación se mencionaran los comandos AT y PID's que se utilizan:

- "AT AR".- Este comando se encarga de reiniciar por completo el escáner ELM327, se utiliza especialmente para evitar que inicie con algún comando anterior o incluso la selección del protocolo de comunicación y de esa manera iniciar desde cero la utilización del escáner.
- "AT SP 00".- Este comando se encarga de seleccionar automáticamente el protocolo de comunicación con el automóvil, de esta manera se puede asegurar que el escáner se puede conectar con distintas marcas de automóviles de una manera automática.
- "010D".- Este comando solicita a la computadora del automóvil (ECU) la velocidad en la que se encuentra el automóvil.
- "010C".- Este comando solicita a la computadora del automóvil (ECU) la velocidad de giro del motor (RMP).
- "012F".- Este comando solicita a la computadora del automóvil (ECU) el nivel de combustible con el que cuenta.

La respuesta que ofrece la computadora del automóvil se debe de interpretar de la siguiente manera:

Por ejemplo se utilizará el parámetro "010C" Este comando mide las RPM del vehículo.

Tabla 3.1 Parámetros de identificación.

Respuesta	Valor	Descripción
"410CAB"	4	El comando fue recibido e indica que esta es la respuesta.
	10C	Indica el comando que se envió y la respuesta del mismo
	A	Primer valor en Hexadecimal.
	B	Segundo valor en Hexadecimal.



Tanto el valor “A” como el Valor “B”, siendo valores hexadecimales es necesario que se les aplique una conversión a un número decimal para ser utilizados en la siguiente formula:

Ecuación 1. PID de RPM

$$RPM = \left(\frac{(A * 256) + B}{4} \right)$$

De esta forma se obtiene la información del vehículo a través del escáner, el resto de los comando cuentan con alguna variante en el número de valores y la ecuación en la que se empleé. El escáner *ELM327* por ser compatible con distintas marcas automotrices, trabaja con parámetros muy generales por lo consiguiente si se requiere utilizar información más específica del algún vehículo se deberá contar con otra clase de escáner.

3.3 Interfaz de comunicación Vehículo - Procesamiento

Una interfaz se encarga de enlazar distintas partes de un sistema, es de suma importancia que se implemente adecuadamente, retomando el diseño conceptual, indica que dependiendo de los módulos a conectar se elige una interfaz de comunicación, los elementos a enlazar son en primera instancia la adquisición de información que se lleva acabo con el escáner *ELM327* y en segundo plano el módulo de procesamiento de información.

El escáner cuenta con la cualidad de tener una conexión vía *USB* o vía *bluetooth*, se trabajó con ambas configuraciones del escáner y sé observo que la comunicación de manera *USB* cuenta con la desventaja de que necesita estar conectado a la *PC* para poder alimentarse eléctricamente, en cambio el escáner de tipo *bluetooth* se alimenta eléctricamente al conectarlo al automóvil.

Después de analizar ambas configuraciones se concluye que por las razones de la alimentación eléctrica además del evitar cables lo cual puede beneficiar al montaje e instalación del sistema en el vehículo, el escáner a utilizar es el vía *bluetooth*, infiriendo que la interfaz de comunicación a utilizar es de tipo inalámbrica específicamente *bluetooth*.

3.3.1 Comunicación bluetooth

La comunicación *bluetooth* posibilita la transmisión de datos mediante un enlace de radiofrecuencia y los objetivos principales son:

- Facilitar las comunicaciones entre dispositivos móviles y fijos.
- Eliminar cables y conectores.
- Ofrecer la posibilidad de crear pequeñas redes inalámbricas.

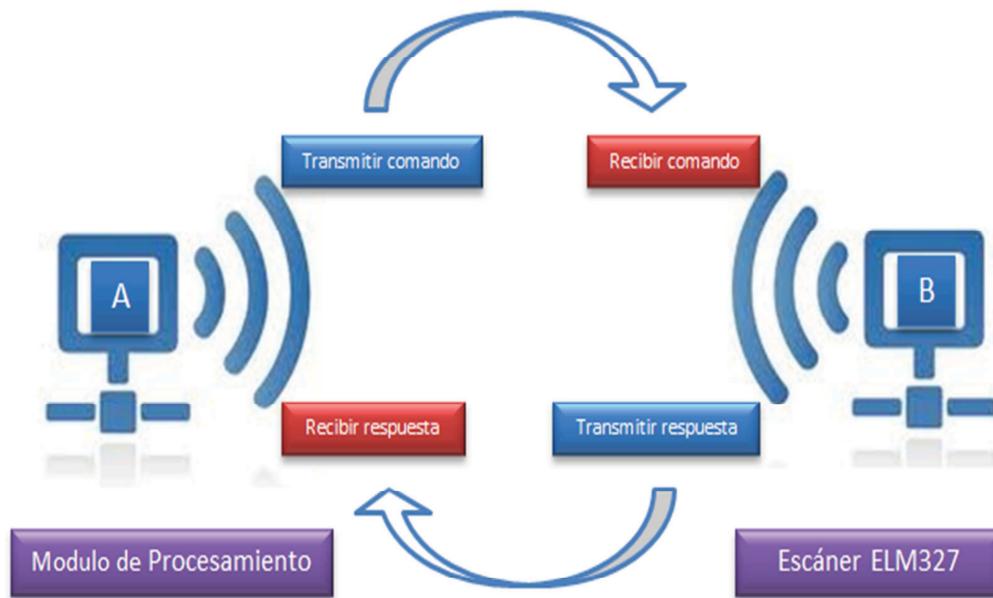


Diagrama 3.1 Comunicación Bluetooth

El elemento "A" se encargará de transmitir el comando, esto se logró con el apoyo de una antena *bluetooth* adaptable al módulo de procesamiento. El elemento "B" recibe el comando y lo comunica a la *ECU* generando una respuesta que posteriormente se envía al elemento "A" el cual la recibe para un procesamiento posterior.

Existen gran variedad de módulos *bluetooth*, que van desde simples antenas hasta módulos muy sofisticados, para este caso se implementará un módulo *bluetooth* de gama media, la razón por la que se elige este dispositivo es la facilidad de configurar ciertos parámetros, como la velocidad, el modo en que trabaja, etc. Con esta ventaja podemos sincronizar sin problemas a la antena que se encuentra dentro del escáner *ELM327*. El módulo seleccionado es el *bluetooth modem - BlueSMiRF Gold* de la marca *Sparkfun®* [13].

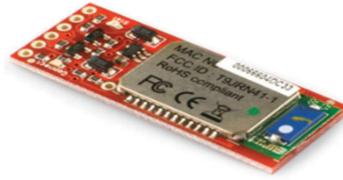


Ilustración 3.3 Antena Bluetooth BlueSMiRF, fuente www.sparkfun.com

Este dispositivo cuenta con una configuración de fábrica ya con ciertos parámetros preestablecidos, a continuación se muestra la tabla donde se indica los parámetros modificados para la implementación de este módulo Bluetooth en este sistema:

Tabla 3.2 Parámetros de Bluetooth

Parámetros	Configuración de fábrica Antena Bluetooth	Configuración Ideal Antena Bluetooth	Escáner ELM327
BT Address	00:06:66:43:40:76	00:06:66:43:40:76	00:19:5D:24:B7:63
BT Name	RN42-4076	RN42-4076	OBD-II
Buad Rate	115K	9600 (1)	9600
Parity	None	None	None
Mode	Slaw	Master 2 (2)	Slaw
Pin Code	1234	1234	1234
A Remote	-	00:19:5D:24:B7:63 (3)	-

Los principales cambios son:

- 1) La velocidad en baudios, es un parámetro muy importante en la configuración del módulo, ya que es de suma importancia que ambos dispositivos a conectar cuenten con la misma velocidad para que no exista pérdida de información. El escáner *ELM327* cuenta con una velocidad predeterminada la cual no se puede alterar, por esa razón se modifica la velocidad del módulo bluetooth de 115k a 9600 baudios.
- 2) El modo en cual trabaja el modulo *bluetooth*, existen distintas modalidades de funcionamiento, para este dispositivo el cual se encuentra preestablecido en modo esclavo indicando que puede ser descubierto y conectarse con ayuda de otro dispositivo *bluetooth*. El modo maestro 2 fue el indicado para la conexión con el escáner, el modo maestro 2 actúa de tal manera que se conecta automáticamente con la dirección remota que tenga predeterminadamente en su memoria.



- 3) La dirección remota, es un parámetro que se puede aprovechar en base al modo de trabajo del *bluetooth*. El escáner trabaja en modo esclavo necesitando un maestro que indique el código de emparejamiento para iniciar la comunicación. Como no se puede realizar la conexión manual entre ambos dispositivos se recurre a guardar previamente el *address* del escáner en la dirección remota del módulo de *bluetooth* para sacar ventaja con su modo de funcionamiento.

Tabla 3.3 Parámetros de comunicación (escáner – módulo bluetooth)

Parámetro	Valor
Bits por segundo	9600
Bits de datos	8
Paridad	Ninguno
Bits de parada	1
Control de flujo	Ninguno

Con lo anterior se tiene establecida la conexión entre ambos elementos para realizar la transmisión y recepción de datos, a la salida de la antena *bluetooth* se tiene una serie de pines que ayudan a la conexión con el módulo de procesamiento de información:

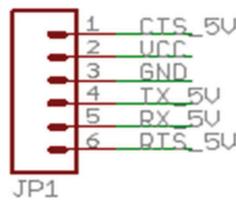


Diagrama 3.2 Conexión del bluetooth

El tipo de configuración que se utilizará para la conexión entre el dispositivo *bluetooth* y el módulo de procesamiento es del tipo “No Handshake”, en el diagrama 3.3 se representa su conexión.



Diagrama 3.3 Conexión no-handshake



La interfaz de comunicación vehículo – procesamiento, transmite los comandos y respuestas necesarios para monitorear los parámetros del automóvil que indican la información que se mostrara en el tablero de instrumentos. En conclusión el siguiente diagrama resume toda la interfaz de comunicación:



Diagrama 3.4 Interfaz vehiculó - procesamiento

3.4 Módulo de Procesamiento de información

El módulo de procesamiento de información es una de las partes más importantes del sistema ya que en él se implementa el algoritmo necesario para realizar de la comunicación con el vehículo y la pantalla. Un algoritmo es un conjunto prescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos.

El manejo de la pantalla por sus características dificulta la selección de módulo que realizará el procesamiento, unos de los parámetros importantes es la manera se comunicara la pantalla considerando la velocidad de transmisión de datos. En esta etapa del proyecto ya se conoce la pantalla que se utilizara, dicho elemento cuenta con un controlador grafico que permite una comunicación de tipo serial facilitando la utilización de un microcontrolador para que se encargue de la etapa de procesamiento de la información.

Durante el diseño conceptual se mencionaron distintas opciones para realizar el procesamiento de información, para el modelo funcional se utiliza el microcontrolador *ATmega2560* contenido dentro de la placa *Arduino Mega®*, este dispositivo es un sistema de desarrollo que se utiliza para proyectos donde se requiere un modelo de manera rápida y fácil, la placa es una adaptación para el microcontrolador donde se acondicionan las entradas y salidas facilitando su conexión y programación [14].



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO



Las principales razones por la cual se seleccionó este dispositivo fue el conocimiento previo que se tenía sobre él, la facilidad para realizar el algoritmo que efectuará toda la lógica del sistema, además de contar con múltiples puertos seriales para su comunicación tanto con el automóvil como con la pantalla.



Ilustración 3.4 Placa Arduino Mega 2560, fuente www.arduino.cc/es

El software que se utiliza para efectuar el algoritmo lleva el nombre arduino, en la versión 1.0, este mismo está basado en *processing* con una serie de librerías que complementan su trabajo. El software Arduino entre sus múltiples librerías cuenta con una que se especializa para comunicarse con *displays* o más específicos pantallas *OLED* de ciertas marcas, se aprovecha esta virtud para su utilización dentro del proyecto. El algoritmo que se implementará para el procesamiento de información se sintetiza en el siguiente diagrama de flujo.

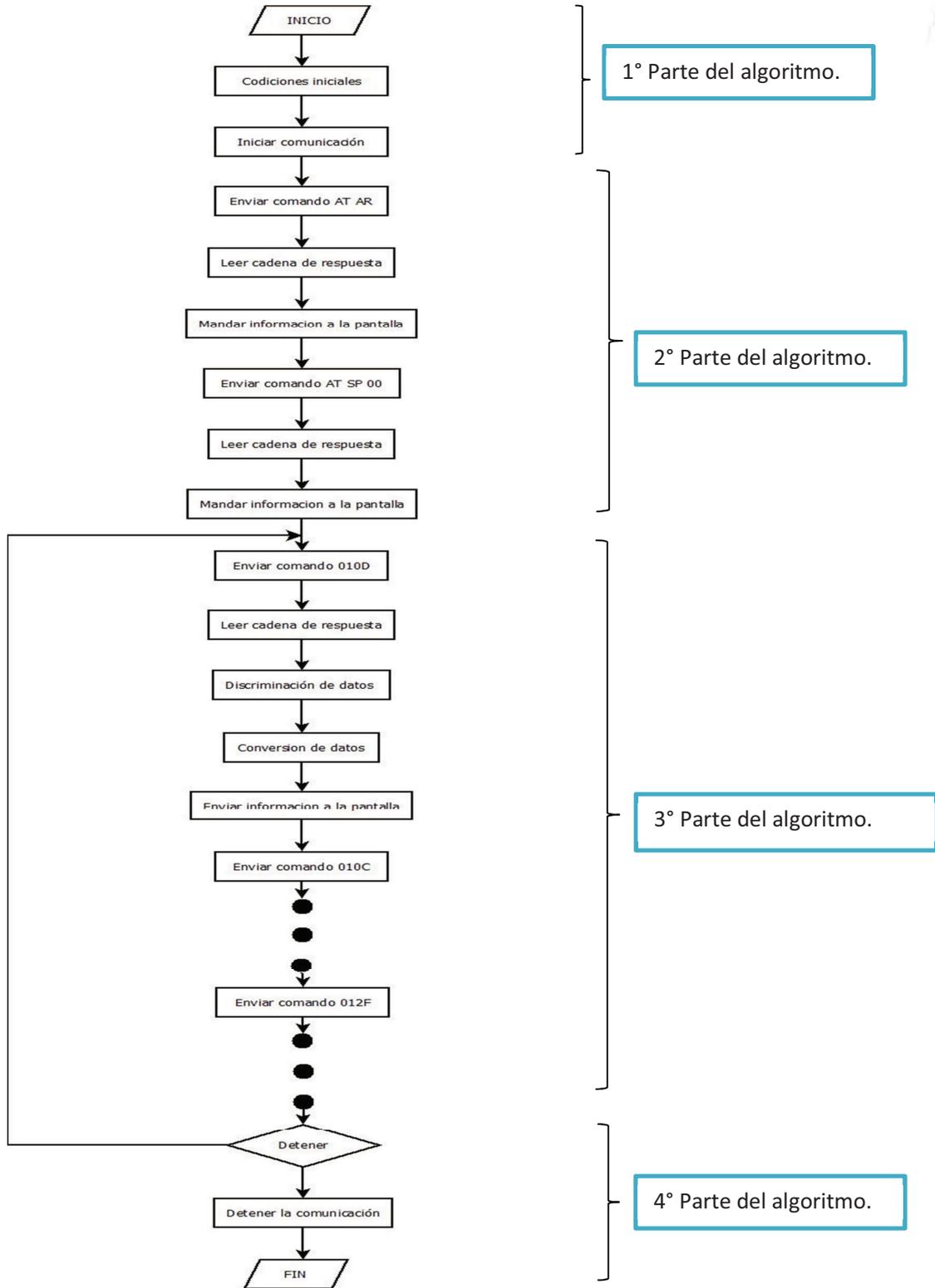


Diagrama 3.5 Algoritmo general



Descripción del Algoritmo:

3.4.1 1° Parte del algoritmo

El primer paso es establecer las condiciones iniciales del algoritmo. Al iniciar se necesita añadir las librerías que se ocuparán, en este caso únicamente se aumentará la librería que se encarga de controlar la pantalla *OLED* que lleva por nombre “<displayshield4d.h>” además de las librerías predeterminadas que tiene el mismo software arduino.

Posteriormente se declaran todas las variables con su respectivo tipo ya sea entero, flotante o carácter, de igual forma los arreglos de variables o constantes que se utilizarán durante el algoritmo. El siguiente paso es definir las funciones específicas, en este punto se enfoca principalmente a unas funciones de inicialización de la pantalla *OLED* y limpieza para evitar cualquier tipo de información visualizada en la pantalla que no sea deseada, con esto se puede asegurar que la pantalla se encuentre reiniciada.

Por último en esta etapa del algoritmo se encarga de configurar los puertos seriales de la placa arduino indicando que el “Puerto0” se ocupará para la comunicación con la pantalla *OLED* y el “Puerto1”, para la comunicación con el escáner. Los parámetros que se configuran es la velocidad de transmisión quedando establecida en 9600 baudios para ambos puertos además de realizar una limpieza a los mismos puertos para evitar la existencia de datos anteriores no deseados y asegurarse que los puertos estén listos para su funcionamiento.

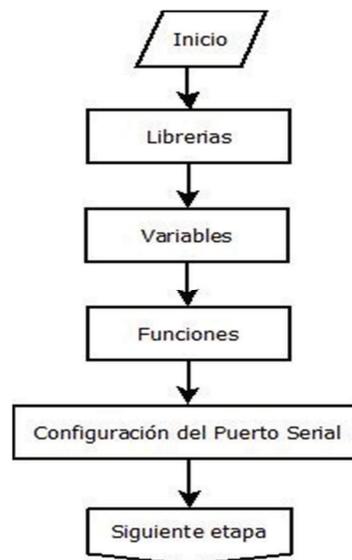


Diagrama 3.6 1° Parte del algoritmo



3.4.2 2º Parte del algoritmo

La segunda etapa del algoritmo básicamente es configurar la comunicación con la computadora del automóvil, enviando comandos AT como el “AT AR” que se encarga de reiniciar la comunicación con el escáner y el comando “AT SP 00” que su función consiste en seleccionar automáticamente el protocolo de comunicación, dependiendo de la marca del automóvil, obteniendo como resultado establecer la comunicación directamente con la *ECU* a espera de la solicitud de parámetros de identificación.

Como se denota en el diagrama lo primero en realizar es enviar el comando “AT AR” por el Puerto1, posteriormente la computadora del vehículo se encargará de verificar este comando y enviar la respuesta que quedará guardada en memoria a través de un arreglo de datos establecido en las condiciones iniciales. Lo que continua es utilizar el comando de control de la pantalla *OLED* que se enviará por el Puerto0, para este caso se empleará un comando que visualice un texto de bienvenida por un tiempo determinado para posteriormente utilizar el comando que se encarga de limpiar la pantalla para dar paso al siguiente comando “AT SP 00” que se trabaja de manera similar al primero.

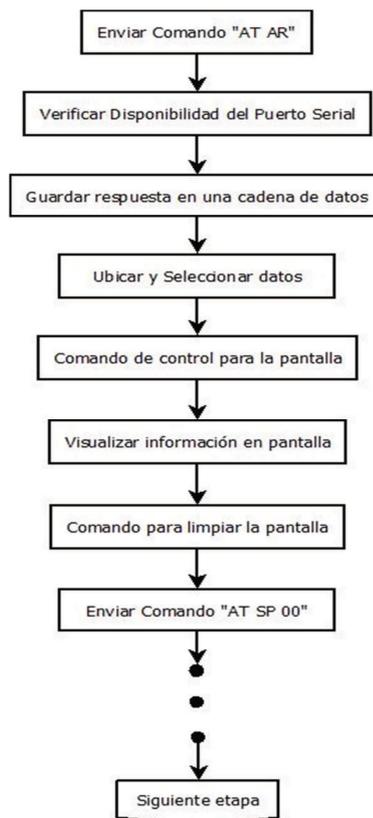


Diagrama 3.7 2º Parte del algoritmo



3.4.3 3° Parte del algoritmo

La tercera etapa es de suma importancia puesto que en ella se realiza la obtención de la información proveniente del automóvil que se mostrará en la pantalla para el usuario. Es primordial que la información presentada se encuentre actualizando de manera constante, provocando que se solicite a la *ECU* múltiples veces, se decide incluir esta etapa en un ciclo infinito que se mantendrá durante toda la operación del sistema.

Lo que ocurre dentro de ciclo de la tercera etapa es muy similar a lo que sucede con la etapa 2. Se inicia enviando el comando "010D" que interpretara la computadora del automóvil como el parámetro de velocidad del vehículo posteriormente la *ECU* envía su respuesta en forma de una cadena hexadecimal.

La cadena recibida queda guardada en un arreglo de datos, ésta cuenta con datos que indican que es respuesta además de especificar el comando que se solicitó, por esa razón se deben de seleccionar solo los valores que sirven para obtener el valor del parámetro deseado, una vez ubicado estos datos se les realiza una conversión a decimal, para poder incluirlos en la ecuación correspondiente ya que esta varía dependiendo del *PID*.

Lo siguiente será representar esta información en la pantalla con ayuda de comandos propios de la librería "<displayshield4d.h>" cuya función consiste en la selección de imágenes que representaran al valor obtenido del automóvil. La imagen se mantendrá durante un tiempo determinado hasta que se vuelva a solicitar nuevamente el comando.

Este proceso es similar para los comandos restantes "010C" y el "012F", RPM y nivel de combustible respectivamente, al terminar de visualizar la información referente al comando "012F" se emplea nuevamente el comando de limpieza de la pantalla para poder mostrar una actualización de la información deseada, toda la etapa sucede rápidamente para conocer las condiciones del automóvil prácticamente en tiempo real.

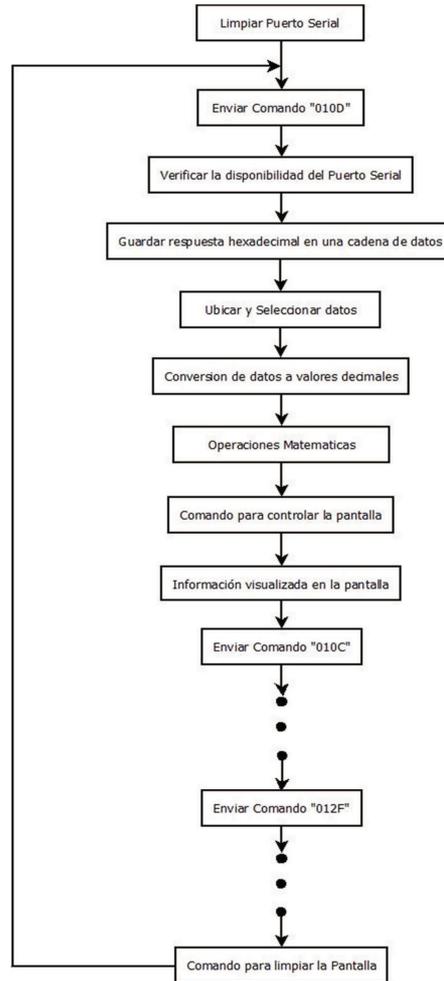


Diagrama 3.8 3º Parte del algoritmo

3.4.4 4º Parte del algoritmo

Básicamente consiste en la finalización del algoritmo, para llegar a esta etapa se tiene previamente que el automóvil se encuentre detenido donde el usuario tendrá la decisión de apagar por completo el automóvil o continuar, en el momento que se encuentre detenido el vehiculó los parámetros se van a ceros si el conductor decide seguir con sus recorrido el algoritmo retorna nuevamente a la 3º etapa y prosigue con la petición de información para procesarla y mostrarla en la pantalla. Si el usuario ha decidido detener el automóvil por completo esto infiere apagar el vehículo por completo, el sistema detiene la comunicación con la ECU principalmente por la falta de energía eléctrica, el sistema también contara con un interruptor propio donde usuario podrá decidir el tiempo en que el sistema trabaje.



El sistema entrar en acción una vez el interruptor se encuentre encendido y que el automóvil se encienda de nuevo reiniciando el algoritmo por completo, para conocer el código que se empleó en el software arduino 1.0 para el sistema de tablero de instrumentos consultar el *apéndice C*.

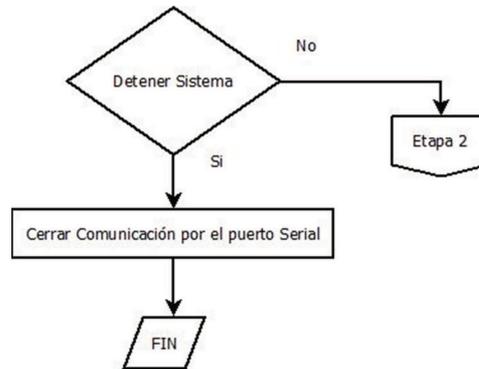


Diagrama 3.9 4º Parte del algoritmo

El módulo del procesamiento de información trabaja como se mencionó anteriormente, este algoritmo ésta enfocado para el arduino mega pero si se utiliza algún otro dispositivo para realizar esta tarea será muy similar sólo con algunas variantes propias de cada dispositivo.

Conforme a la arquitectura del sistema definida en el diseño conceptual la siguiente etapa del proyecto es la interfaz de comunicación entre el modulo de procesamiento en este caso la placa arduino con la pantalla *OLED*.

3.5 Interfaz de comunicación Procesamiento - Pantalla

Enlazar el módulo del procesamiento de información con la pantalla *OLED* dependió de gran manera de la selección de los dispositivos a utilizar, como ya se sabe la parte del módulo de procesamiento será efectuado por el microcontrolador ATmega2560 y la pantalla seleccionada para el desarrollo de un modelo funcional del sistema de tablero de instrumentos es de la marca “4D Systems” empresa Australiana especializa en la producción de *displays* inteligentes y una de sus ramas son las de tipo *OLED*.

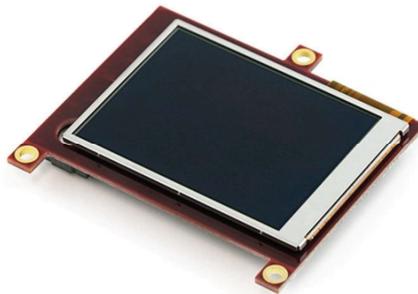


Ilustración 3.5 Pantalla OLED 4D Systems, fuente www.4dsystems.com.au

3.5.1 Controlador gráfico GOLDELOX

Las pantallas de la empresa *4D Systems* ya cuentan con un controlador gráfico que facilita su utilización, los principales son *GOLDELOX* y *PICASO*, el primero se utiliza para pantallas de menor tamaño y la segunda para pantallas de mayor tamaño y táctiles. Este tipo de *displays* son muy versátiles para el desarrollo de prototipos pero cuentan con una gran limitante, el tamaño de las pantallas es muy pequeño teniendo como máximo una pantalla de 3.2 pulgadas.

La pantalla *OLED* que se implementó para el modelo funcional fue una de tipo *PMOLED (Passive Matrix OLED)*, esta pantalla se tendrá que sustituir para un prototipo final, ya que no cuenta con las características contempladas durante el diseño conceptual, en la siguiente tabla se muestra las características de la pantalla seleccionada [15]:

Tabla 3.4 Características de la pantalla OLED

Característica	Valor
Modelo	uOLED 160 G1
Resolución	160x128 pixeles
Color	RGB 65k colores
Transparencia	No
Flexibilidad	No
Largo de pantalla	40 mm
Ancho de pantalla	33.5 mm
Grosor de pantalla	1.5 mm
Controlador grafico	GOLDELOX
Comunicación	Serial



El controlador gráfico *GOLDELOX* tiene la disponibilidad de ser configurado en dos modos de funcionamiento:

- El modo “**GFX**” permite a la pantalla tomar ventaja de su procesador permitiendo que se pueda incluir un algoritmo en el mismo, volviéndolo independiente esto con ayuda del lenguaje de programación *4DGL* propio de la empresa y semejante al lenguaje C.
- El modo “**SCG**” la pantalla toma el papel de esclavo y necesita otro dispositivo que le indique los comandos para su funcionamiento, en este modo de trabajo el procesador gráfico se quede a la espera de órdenes para interactuar con la pantalla. La manera de comunicación de los comandos es vía serial.

EL procesador gráfico no permite utilizar ambas configuraciones al mismo tiempo por esa razón se requiere tener definido el modo de trabajo que operara la pantalla. La forma de configurar el procesador consiste en guardar un archivo *PmmC* en el mismo a través de un software que lo proporciona la misma empresa *4D Systems* que lleva por nombre *PmmC Loader* [16].

Para cargarle el archivo *PmmC* a la pantalla se requiere de un dispositivo complementario que facilita la conexión a una PC. Es un convertidor serial (TTL) a micro USB, su funcionamiento se basa en un integrado *CP2102*, el modelo es *uUSB-MB5* [17]. Las funciones del convertidor no solo se limitan a elegir el archivo *PmmC* que uno desee, además se utiliza en el modo *GFX* para poder programar al procesador gráfico.

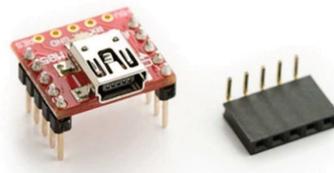


Ilustración 3.6 *uUSB-MB5*, cortesía de www.sparfunk.com

El convertidor *uUSB-MB5* se acopla perfectamente a la pantalla mediante una serie de pines situados en la parte posterior conocido como el conector CN3 del display *OLED*. El conector, también es utilizado para conectar de manera serial a un dispositivo que trabaje como maestro permitiendo enviar los comandos necesarios para el control de la pantalla, esto cuando se trabaja en el modo *SGC*.

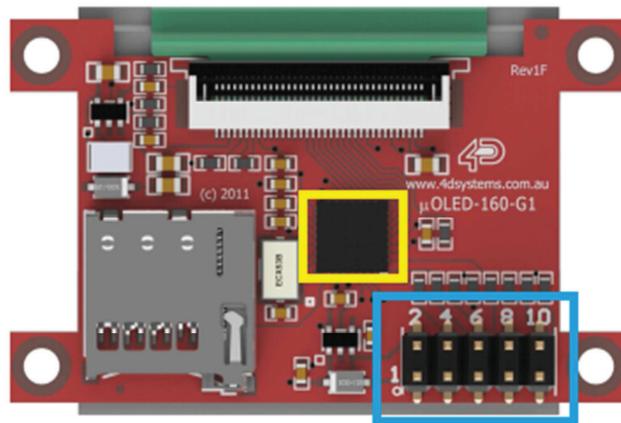


Ilustración 3.7 Pantalla uOLED 160G1, cortesía de www.4dsystems.com.au

- Procesador gráfico *GOLDELOX*.
- Conector CN3.

Tabla 3.5 Conector CN3

Conector CN3		
PIN	Símbolo	Descripción
1	Vcc	Voltaje de alimentación principal 5V nominales.
2	NC	No conectar.
3	Tx	Transmisión asíncrona serie, datos en niveles TTL.
4	IO2	Entrada o Salida 2 de propósito general.
5	Rx	Recepción asíncrona serie, datos en niveles TTL.
6	IO1	Entrada o Salida 1 de propósito general.
7	GND	Tierra.
8	GND	Tierra.
9	RESET	Pin de entrada para reinicio de señal.
10	3.3 Vout	Voltaje de salida 3.3V, 50mA.

Se define que el modo de trabajo de la pantalla es el SGC teniendo como maestro a la placa arduino mega entablando una comunicación de tipo serial. El maestro se encargara de mandar las instrucciones necesarias contenidas en el algoritmo y el procesador gráfico las interpretará y las ejecutará.

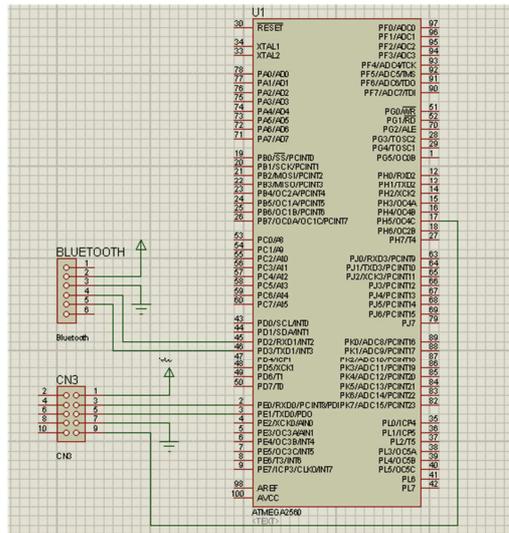


Diagrama 3.10 Conexión pantalla - ATmega2650 – Antena Bluetooth

En resumen la conexión con el microcontrolador tanto con el *bluetooth* como con la pantalla se establecerá como se muestra en el diagrama 3.10, cabe resaltar que se deberá consultar los diagramas de la placa arduino mega, para conocer que número de pin correspondiente al microcontrolador, la alimentación Vcc y la tierra que se conectan a la pantalla y al *bluetooth* se puede tomar de la misma placa arduino, ya que el consumo de corriente se encuentra dentro del rango permitido para la placa.

3.6 Visualización de la información en la pantalla

En la etapa anterior se dio a conocer la forma de configurar y conectar la pantalla *OLED* al módulo de procesamiento, en la parte de proyección de la información consiste básicamente la manera en que se mostrará la información al usuario con las instrucciones necesarias para la interacción con el mismo *display*.



Ilustración 3.8 Imagen en pantalla OLED, fuente www.4dsystems.com.au



Cuando se trabaja la pantalla OLED en el modo SGC se controla a través de una serie de comandos enviados por comunicación RS232 desde el dispositivo maestro. Estos comandos ya se encuentran establecidos en un manual propio de la empresa 4D Systems. Existen aquellos que representan en el display un punto, una línea, un polígono, alguna imagen o video, textos, limpiar la pantalla, etc. La finalidad de ellos es implementarlo con respecto a las necesidades del proyecto en desarrollo. El procesador gráfico GOLDELOX se encargara de procesar todos estos comandos e interactuar con la pantalla

3.6.1 Comandos de control para pantalla

El manual de la empresa 4D Systems explica el manejo de comandos además complementa con documentos la comprensión del modo de trabajo de la pantalla, para conocer los comandos detalladamente se sintetiza la información de uno de ellos en una tabla para entender su funcionamiento y aplicación [18], para mayor detalle ver apéndice C:

Tabla 3.6 Descripción de un comando para visualizar un texto en pantalla OLED

Comando para Visualizar un texto.	
Comando	String(cmd,column,row,font,stringColour,"string", termination)
Cmd	73(hex) o s(ascii): Byte de comando de cabecera.
Column	Posición inicial horizontal del texto: Rango depende del tamaño de la pantalla.
Row	Posición inicial vertical del texto: Rango depende del tamaño de la pantalla.
Font	Byte que especifica la fuente interna que utilizara el texto, tipo de letra, proporción etc. Ejemplos: 0: 5x7 fuente interna 1: 8x8 fuente interna 2: 8x12 fuente Interna Esto byte puede ser alterados y añadir otras fuentes.
StringColour	2 Bytes que define el color del texto.
String	Texto a representar como máximo 256 caracteres.
Terminator	Byte indica que el texto terminado con 00(hex)
Respuesta	Acknowledge
	06(hex) ACK byte indica que fue exitosa la comunicación. 15(hex) NAK byte indica que fue errónea la comunicación.
Descripción	Este comando representará un texto, en cualquier lugar de la pantalla indicando la posición inicial horizontal y vertical, indicando el fin del comando con un 00(hex), la longitud máxima del texto no deberá exceder los 256 caracteres. <div data-bbox="1018 1518 1385 1765" data-label="Diagram"> </div>
Ejemplo	Command Data: (73hex, 07hex, 08hex, 01, FFhex, FFhex, "Hola", 00hex) Muestra: "Hola" iniciando en el punto (7,8) fuente 8x8, de color blanco.



La aplicación de pantallas de la marca *4D Systems* con las placas arduino es muy común, por consecuencia algunos programadores desarrollaron librerías contemplando los comandos más comunes de la pantalla para poder condensar el código y facilitar su utilización. La librería de arduino que se utilizó lleva por nombre “*displayshield4d*” desarrollado por Oscar González teniendo como última actualización a principios del año 2011 [19].

La librería contiene procesos esenciales como el de inicio de comunicación, limpiar la pantalla, escribir textos, dibujar líneas, rectángulos, círculos, triángulos, etc. Lo primero es inicializar la comunicación con la pantalla y limpiar la misma en las condiciones iniciales del algoritmo, asegurando un reseteo por completo. La sintaxis de estos comandos queda representada en el software arduino como:

- **Oled.Init();** Inicializa la pantalla.
- **Oled.Clear();** Limpia lo que se encuentre en la pantalla.

La siguiente parte donde se utilizará la pantalla en el algoritmo es al presentar un texto que indique el inicio del funcionamiento del tablero de instrumentos, durante la etapa 2 se implementa dicho texto que indica una bienvenida al usuario y otro texto indicando el inicio del trabajo. La librería ya contempla una función que representa un texto, similar al mencionado en la tabla 3.6, la sintaxis que se utilizará en el software arduino para la utilización de la función queda de la siguiente manera:

- `drawstringblock(x,y,font,stringColor,width,height,“Text”);`
- `drawstringblock(20,20,0,oled.RGB(255,255,255),1,1,“BIENVENIDO”);`

En resumen se presentará el texto “BIENVENIDO” iniciando en la columna 20 y fila 20 de la pantalla, con una fuente preestablecida, con un largo y ancho del texto de 10 ya que el valor escrito en la sintaxis está multiplicado por 10, para el segundo texto es similar. Básicamente esta es la metodología a utilizar en el software arduino para interactuar con la pantalla *OLED*.

3.6.2 Memoria uSD en pantalla OLED

La representación de la información de la velocidad, RPM, y nivel del combustible de automóvil no se representara con un texto sino con una serie de imágenes, para realizar esta tarea se tomara ventaja de una cualidad de la pantalla, aquella que permite adaptarle una memoria de tipo *uSD* donde se puede guardar imágenes, videos, y en algunas pantallas algún archivo de audio y posteriormente con algunos comandos poder mostrarlas en la pantalla.

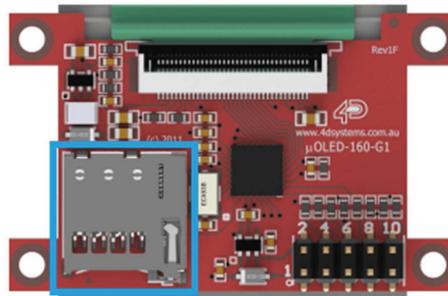


Ilustración 3.9 uSD pantalla OLED, fuente www.4dsystems.com.au

● Entrada memoria uSD.

Implementar una memoria *uSD* en la pantalla *OLED*, se debe de considerar algunos aspectos previos, dependiendo de la pantalla a utilizar la capacidad de memoria varía, la pantalla *uOLED 160 G1* tiene como máximo 2 Gb. La memoria *uSD* se debe de configurar en un formato específico, la empresa recomienda crear dos particiones de la memoria una en formato *FAT32*, y la segunda en formato *RAW*. La manera en que se consigue configurar la memoria en estos dos formatos es con un software llamado *RMPET-Tool* propio de la empresa *4D Systems* [18].

El software *Graphics Composer* se encarga de guardar las imágenes y videos en la memoria *uSD* proporcionando 3 valores (*h,m,l*) que indican la localidad de memoria donde se encontrara el archivo y con esa información poder utilizarse cuando se requiera. Los software utilizados se encuentran incluidos en el paquete “*4D Workshop 3 IDE*” que puede ser descargado gratuitamente de la pagina oficial de la empresa Australiana [20].

La librería *displayshield4d* no contempla alguna función para poder visualizar las imágenes directamente de la memoria *uSD*, pero retomando el manual de comandos además de textos complementarios se puede conocer lo que se requiere para dicha función e implementarlo en la librería, definiendo la sintaxis de la siguiente forma:

- `ScrCopyDis(x,y,width,height,h,m,l);`

Ejemplo considerando que la imagen se visualizará con una resolución de 160x128 pixeles y se encuentra guardada en la localidad *h=0x0F*, *m=0x2A*, *l=0x00*, la sintaxis se establecería de la siguiente forma:

- `ScrCopyDis(0x00,0x00,0xA0,0x80,0x0F,0x2A,0x00);`



Con la función anterior se puede mostrar una serie de imágenes que representen la información de velocidad, RPM, nivel de combustible. El siguiente paso contempla la organización de la información dentro de la pantalla para esto es necesario otorgar un espacio específico en pixeles a cada elemento considerando que la información visualizada sea clara, legible y entendible para el usuario.

3.6.3 Representación gráfica del tablero de instrumentos para vehículo en pantalla OLED.

La pantalla *uOLED 160 G1* cuenta con una resolución de 160x128 pixeles donde se representara toda la información adquirida del vehículo, la cual estará actualizando de manera constantemente para conocer el estado del automovil, en cambio existe información que no sufrira cambio alguno mientras el sistema se encuentre trabajando por ejemplo el simbolo del kilometraje, algun texto, etc.

Es de suma importancia asignar el espacio requerido de la pantalla a la información que cambien además de la que se quede fija durante su funcionamiento. El siguiente diagrama indica en que manera estará organizada la informacion dentro de la pantalla.

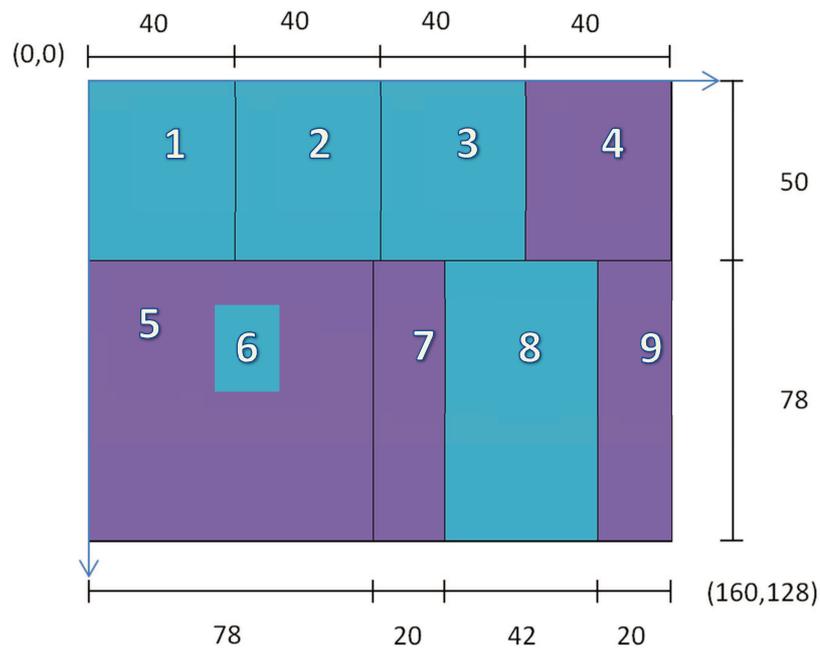


Diagrama 3.11 Distribución de la información en la pantalla OLED

- Información variable.
- Información fija



- 1.- Centenas de la velocidad (0,1,2).
- 2.- Decenas de la velocidad (0 a 9).
- 3.- Unidades de la velocidad (0 a 9).
- 4.- Imagen fija de Km/h.
- 5.- Imagen fija de tacómetro.
- 6.- Aguja indicadora de RPM.
- 7.- Imagen fija de RPM.
- 8.- Indicador de nivel de gasolina en %.
- 9.- Imagen fija de FUEL.

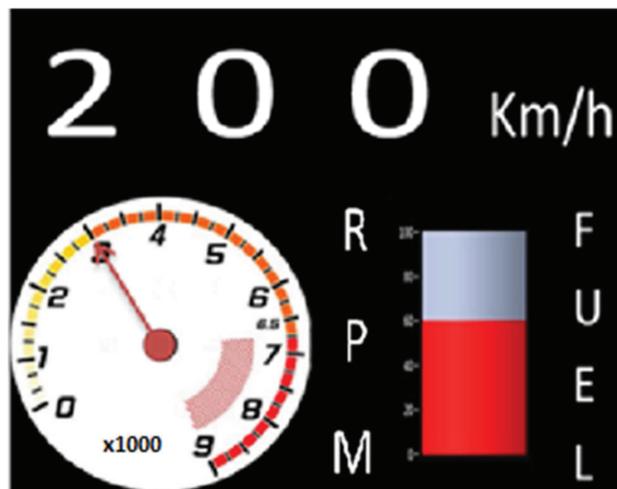


Ilustración 3.10 Tablero de instrumentos

En resumen todo lo anterior es la metodología a utilizar para la visualización de la información en la pantalla *OLED*, con algunas variantes dependientes de las características de la pantalla, como por ejemplo la transparencia donde se deberá considerar el color de la información. La última parte del sistema es el montaje al automóvil, buscando la mejor opción para colocar el modelo funcional del sistema de tablero de instrumentos para vehículo con pantalla *OLED*.

3.7 Montaje del sistema en el automóvil

El montaje del sistema considera varios aspectos, principalmente la posición en la que se ubicará el modelo funcional además de la pantalla *OLED* que muestre la información al usuario. Para definir la ubicación de los elementos se retoma los requerimientos que se establecieron desde el principio del diseño.

3.7.1 Ubicación de la pantalla *OLED* en el automóvil.

Los requerimientos que se deben de considerar para esta etapa son:

- El sistema visualizará la información del tablero en el parabrisas.
- El sistema permitirá al usuario contar con una buena visibilidad del camino.
- La información se visualizará de una manera clara, legible y entendible

Para la selección de la ubicación donde se encontrará la información se realizó un estudio de varias posibilidades de posición:

Tabla 3.7 Comparativa de configuraciones.

	1	2	3	4	5
Posiciones	Izquierda- superior	Centro-inferior	Centro-medio	Centro-superior	Derecha-superior
Visibilidad del camino	Permite la visibilidad del camino	Permite la visibilidad del camino	Impide la visibilidad del camino	Permite la visibilidad del camino	Permite la visibilidad del camino
Visibilidad del camino al ver el tablero de instrumentos	La mirada se inclina un poco horizontalmente no se pierde el camino esta configuración es muy utilizada para medidores de otros parámetros en el automóvil.	La dirección es la misma que con el tablero convencional, se ahorraría distancia y tomaría menos tiempo monitorear los parámetros, siendo esta la más utilizada por los sistemas HUD.	Impide al conductor ver el camino poniendo en riesgo la integridad del operario.	La mira se inclina en dirección superior perdiendo por un leve instante la vista del camino.	La mirada se inclina poco horizontalmente no se pierde el camino pero podría generar confusión con el espejo retrovisor.
Desviación de la mirada	Mínima	Mínima	Nula	Moderada	Mínima
Visibilidad de la información	Si	Si	Si	Si	Si
Montaje y conexión	Encontrándose a los extremos del parabrisas facilitaría su montaje	Encontrándose a los extremos del parabrisas facilitaría su montaje	Encontrándose frente al conductor dificultaría su montaje	Encontrándose a los extremos del parabrisas facilitaría su montaje	Encontrándose en el centro del parabrisas dificultaría su montaje

Se optará por las configuraciones, izquierda-superior y centro-inferior, aprovechando su posición para las conexiones además que ambas son las más recurridas para los sistemas HUD y monitores de parámetros

3.7.2 Conexiones del modelo funcional

Al modelo funcional se decidió incluirle una batería que funcionara como fuente de alimentación propia, la idea original es que el sistema se alimente eléctricamente directamente del automóvil. Se inclinó por utilizar una batería alcalina de 9 [V] con un regulador a 5 [V] que por general cuenta con una carga eléctrica de 1000 [mA].

Las pantalla OLED requiere de 40[mA], el módulo bluetooth trabaja con 25[mA] y por ultimo la placa arduino mega requiere de 50[mA], donde un consumo eléctrico del dispositivo en general es de 115[mA] esto se comprobo prácticamente y teóricamente con ayuda de las hojas de especificaciones de la placa arduino, la pantalla OLED y módulo bluetooth.

Se realizó el calculo el tiempo de la bateria y se concluyo:

Ecuación 2 Tiempo de descarga

$$\text{Tiempo de descarga} = \frac{\text{carga eléctrica de la bateria}}{\text{consumo eléctrico del dispositivo}}$$
$$\frac{1000 [mA]}{115[mA]} = 8.7$$

Esto indica que el sistema trabajar 8.7 horas continuamente, indicando que para una segunda version del sistema se deberá trabajar en su modo de alimentacion. Una vez considerando la alimentacion eléctrica, las conexiones del modelo funcional quedaron de la siguiente manera.

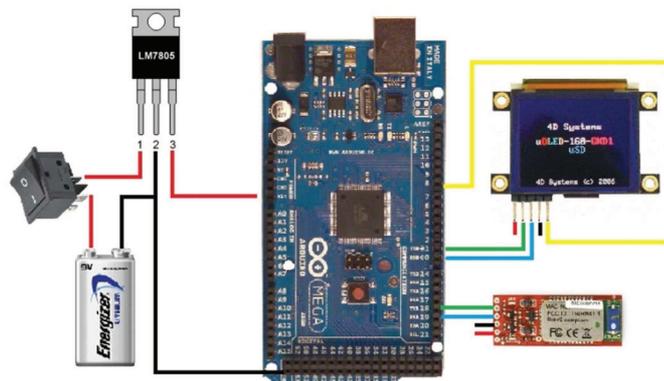


Diagrama 3.12 Conexión del modelo funcional



Se desarrolló un circuito impreso adaptable al arduino mega, en la cual pudiese conectar la pantalla *OLED*, la antena *bluetooth*, además de lo que se necesitara para la alimentación eléctrica y el interruptor del mismo tablero de instrumentos. El circuito se diseñó en el software *EAGLE* y con el apoyo de una maquina CNC para el desarrollo de prototipos de circuitos impresos [21]. Una vez concluido el circuito impreso se logró ensamblar todo el sistema obteniendo como resultado lo mostrado en la ilustración 3.11.

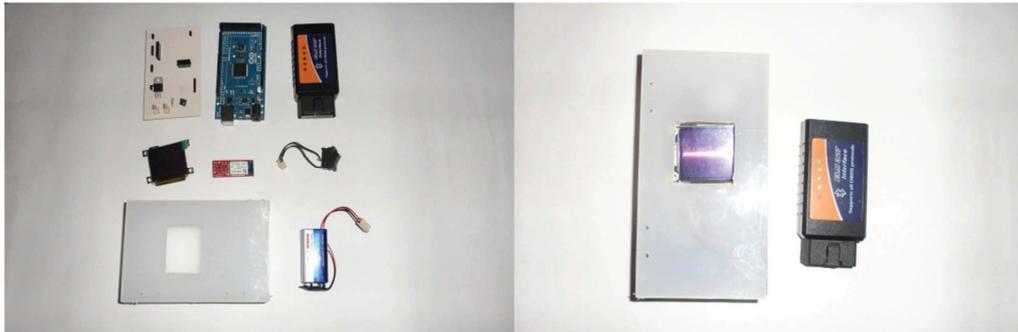


Ilustración 3.11 Sistema separado y ensamblado

3.7.3 Diseño final del modelo funcional CAD

Para una mejor comprensión del modelo funcional se apoyó del diseño asistido por computadora (CAD) con el software *SolidWorks®* donde se dibujaron todos los componentes del sistema contando con planos en el *apéndice E*. obteniendo como resultado el modelo de la ilustración 3.11.

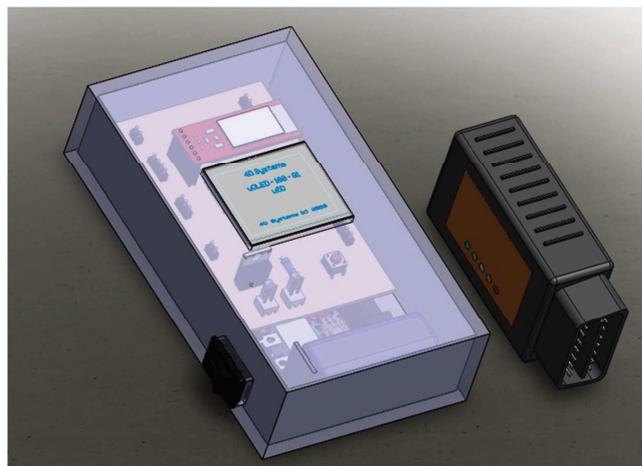
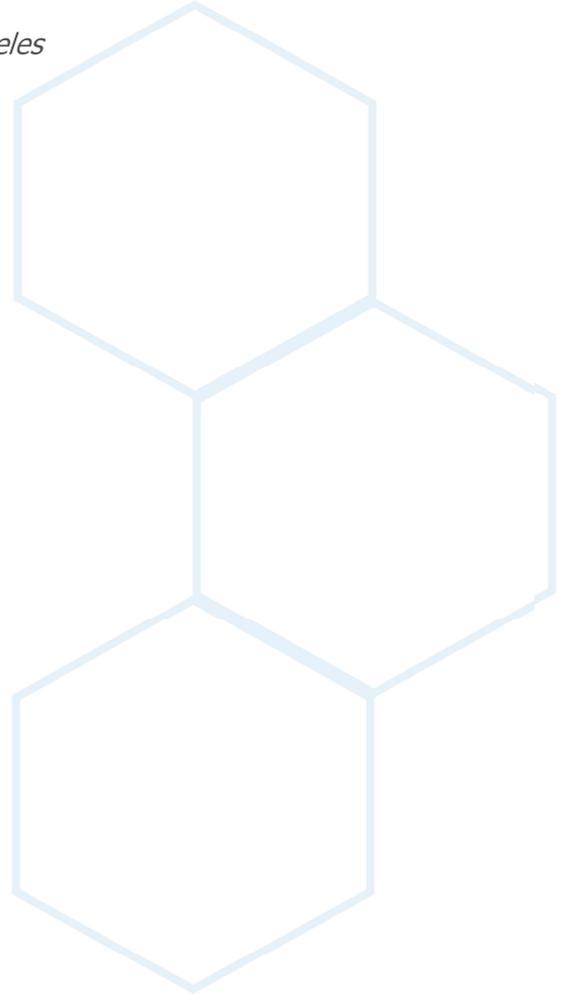


Ilustración 3.12 Modelo funcional (CAD)

Capítulo 4

"La inteligencia consiste no sólo en el conocimiento, sino también en la destreza de aplicar los conocimientos en la práctica"

-Aristóteles



4.1.- Pruebas en laboratorio (PC).

4.2.- Pruebas en campo (Automóvil).





Para corroborar el funcionamiento del modelo funcional es necesario validar aquellas características mínimas de operación. Durante la elaboración y al finalizar el sistema se sometió a distintas pruebas esto con la finalidad de detectar anomalías que provocasen algún cambio en el desarrollo del modelo funcional. Se comprobó el funcionamiento del sistema en general además de cada uno de los elementos que componen el sistema por separado, iniciando con el escáner *ELM327* hasta la el circuito impreso que se diseñó para contener la pantalla.

Al sistema de tablero de instrumentos en general fue sometido a dos pruebas principales:

- 1° Pruebas en laboratorio (*PC*)
 - ✓ Simular el comportamiento de un automóvil con la *PC*.
 - ✓ Descripción de la prueba en laboratorio (*PC*).
 - ✓ Resultados de las pruebas en laboratorio.

- 2° Pruebas en campo (Automóvil)
 - ✓ Automóviles a utilizar para pruebas del sistema.
 - ✓ Descripción de la prueba en campo (Automóvil).
 - ✓ Resultados de las pruebas en campo.

4.1 Pruebas en laboratorio (*PC*)

4.1.1 *Simular el comportamiento de un automóvil con la PC*

Durante el desarrollo del sistema no en todas ocasiones se contó con un automóvil para probar los avances, fue necesario buscar una alternativa para realizar las pruebas pertinentes sin tener que contar con el vehículo. Se utilizó de apoyo una *PC* para simular el comportamiento de la computadora del automóvil. Con ayuda del software *LabVIEW* se desarrolló un programa que se encuentra basado en la comunicación RS232 que permitiera recibir y enviar cadenas de datos, similar a lo que realiza la ECU con los parámetros de identificación a través del escáner *ELM327*.

Retomando lo mencionado en el desarrollo del sistema la comunicación con el automóvil será de tipo RS232 de transmisión inalámbrica (*bluetooth*), para realizar la prueba de simulación con la *PC* es de suma importancia apearse lo más posible al sistema original, para esto se necesitó adaptar una antena *bluetooth* de tipo *dongle* a la *PC* permitiendo una comunicación inalámbrica.



Ilustración 4.1 Bluetooth dongle

Permitiendo una comunicación de tipo inalámbrica se desarrolló un algoritmo en *LabVIEW* que se encarga de entabla una comunicación serial con el modelo funcional, asignaron parámetros iniciales, como son velocidad de transmisión, los bits de parada, etc. Posteriormente se dedica a leer los comandos “AT AR”, “010D”, etc., que son enviados desde el sistema de tablero de instrumentos.

El programa en *LabVIEW* se encarga de identificar el comando y genera una respuesta semejante a la que envía la computadora del automóvil, dicha respuesta puede ser modificada al gusto del operario simulando el cambio de parámetros que sufre el automóvil durante su funcionamiento dígame velocidad, RPM y nivel de combustible. Verificando que el modelo funcional interpreta la respuesta y realiza los cambios pertinentes para mostrar al usuario la información final a través de la pantalla *OLED*.

La finalidad es realizar esta prueba es generar un panorama similar al que se enfrentara el sistema de tablero de instrumentos a su aplicación final, obteniendo como ventaja el lugar de trabajo y la facilidad de realizar cambios al sistema rápidamente.

4.1.2 Descripción de la prueba en laboratorio (PC)

La prueba en el laboratorio consiste básicamente en conectar de manera inalámbrica el modelo funcional a la *PC* donde se correrá el algoritmo desarrollado en *LabVIEW* aproximadamente por un par de horas, durante este lapso de tiempo se modificarán de manera constante las respuestas enviadas desde el programa de la *PC*, simulando la modificación de los parámetros del automóvil (velocidad, rpm, combustible).

Lo que se pretende corroborar con esta prueba son las características de funcionamiento del sistema, la sincronización entre la *PC* y el modelo funcional, verificar la interpretación de los cambios en la información recibida, además de no presentar alguna traba durante su tiempo de operación.



Ilustración 4.2 Prueba de laboratorio (PC)

4.1.3 Resultados de las pruebas en laboratorio:

Como resultado de utilizar una *PC* en sustitución de un vehículo produjo un acercamiento aceptable al que se someterá el sistema directamente con el automóvil. El sistema de tablero de instrumentos presentó una respuesta favorable obteniendo como principales resultados:

- El sistema entabla la comunicación inalámbrica.
- El sistema envía los comandos.
- El sistema recibe y guarda la respuesta.
- El sistema interpreta los cambios en las respuestas.
- El sistema realiza las conversiones y operaciones pertinentes.
- El sistema despliega la información en la pantalla *OLED*.

Durante las pruebas en laboratorio de igual forma se presentaron algunos percances de sincronización entre la *PC* y el modelo funcional, dando como resultante una la interpretación de respuestas desfasadas, orillando al sistema a mostrar información incorrecta.

Esto se ocasionaba por el manejo de tiempo en retardos en ambas partes. Para la solución de este problema se manejó distintas opciones de tiempos en retardo, buscando una sincronización adecuada para el sistema. Se obtuvo como solución 500 [ms] para cada parámetro y 1.5[s] para una actualización completa de la pantalla. Al manejar este tiempo de sincronización la información se presentó de maneja confiable, respetando los cambios generados directamente del programa en *LabVIEW*.

Durante el tiempo que se encontró en operación el sistema no presento alguna traba que no fuese generada por una perturbación externa, por ejemplo corte en el suministro eléctrico, alto inesperado en el programa de la *PC*, etc. En términos generales el sistema respondió de manera esperada pronosticando altas probabilidades de funcionamiento en las pruebas de campo.



4.2 Pruebas en campo (Automóvil)

4.2.1 Automóviles a utilizar para pruebas del sistema

Comprobar la operación del modelo funcional en una situación real es lo que consiste las pruebas en campo, la finalidad es ubicarlo en condiciones a la que será sometido en su aplicación final. Una vez contando con el modelo final se aplicó la prueba para corroborar su funcionamiento, de igual forma esta prueba se realizó durante el desarrollo del sistema buscando detalles que causeasen algún peso en el sistema para provocar algún cambio. Los cambios que se generaban eran sometidos primeramente a las pruebas del laboratorio.

Lo primero fue buscar automóviles a los cuales conectar el sistema, variando en marca y año, esto con la finalidad de comprobar la compatibilidad del sistema. La siguiente tabla indica los distintos vehículos en los que se probó el sistema durante y al finalizar su desarrollo.

Tabla 4.1 Automóviles para pruebas

Automóvil	Marca	Año
Lancer	Mitsubishi	2006
Focus	Ford	2010
Astra	Chevrolet	2004
Verna	Hyundai	2005
Neon	Dodge	2006

4.2.2 Descripción de la prueba en campo (Automóvil)

La prueba en campo consiste básicamente en conectar el escáner *ELM327* en la entrada *OBD-II* del automóvil, posteriormente dar marcha al vehículo para que el escáner entre en modo de espera para iniciar la comunicación, el modelo funcional se conecta inalámbricamente por medio de la antena *bluetooth* dando inicio a la configuración inicial para entablar la comunicación con la computadora del automóvil.

La siguiente parte de la prueba es encender por completo el automóvil, dando pie al cambio de parámetros como la velocidad, en ese momento se inicia la petición y recepción de información. Lo que se pretende comprobar con esta prueba es el funcionamiento y compatibilidad del sistema.



Ilustración 4.3 Prueba de campo (automóvil)

4.2.3 Resultados de las pruebas en campo:

Implementar el modelo funcional ya en condiciones reales es la mejor opción para comprobar el funcionamiento del mismo, los resultados obtenidos de esta prueba durante las etapas de desarrollo, ayudaron a percibir alguna anomalía o variante que se tenía que tomar en cuenta para el sistema final para así posteriormente realizar los cambios pertinentes.

En todos los automóviles antes mencionados se probó el sistema en alguna etapa del desarrollo del sistema, pero la disponibilidad de los vehículos no siempre fue la misma, la mayoría de las pruebas se efectuaron en el Lancer 2006 de la marca *Mitsubishi*, incluyendo al modelo funcional final.

Al igual que las pruebas en laboratorio el sistema presento resultados tales como:

- El sistema entabla la comunicación inalámbrica.
- El sistema envía los comandos.
- El sistema recibe y guarda la respuesta.
- El sistema interpreta los cambios en las respuestas.
- El sistema realiza las conversiones y operaciones pertinentes.
- El sistema despliega la información en la pantalla *OLED*.

Para cada automóvil en el cual fue probado el sistema presento algunas ventajas y desventajas:



Tabla 4.2 Comparativa entre automóviles

Automóvil	Lancer	Focus	Astra	Verna	Neon
Conexión inalámbrica	Si	Si	No se probó se utilizó USB	Si	No se probó se utilizó USB
Comunicación de datos	Si	Si	Si	Si	Si
Procesamiento de datos	Si	Si	Si	Si	Si
Despliegue de información en la pantalla	Si	Si	No se probó con pantalla OLED	No se probó con pantalla OLED	No se probó con pantalla OLED
Ventajas	El sistema trabajó sin mayor inconvenientes, la cadena de respuestas contenía únicamente los datos indispensables	El tiempo de sincronización podía ser hasta 250 [ms]	El sistema realizó el procesamiento de datos	La cadena de respuesta contenía los caracteres mínimos	Se logró visualizar la información del vehículo en un PC
Desventajas	Se presentaron problemas de sincronización a tiempos menores a 500 [ms]	La cadena de respuesta tendía a ser muy grande mayor a 20 caracteres	El vehículo no manejaba con el parámetro de identificación del nivel de combustible	El sistema se desconectaba constantemente probablemente por averías del automóvil	Se realizaron pocas pruebas en una etapa temprana del proyecto

Lo que se puede concluir de las pruebas en campo es que en términos generales el sistema de tablero de instrumento respondió de una manera esperada, presentado algunos percances como la sincronización que de igual forma se notó durante las pruebas en laboratorio, además que la respuesta que envía la computadora del automóvil no es la misma en distintos vehículos orillando al sistema a un reajuste para cada automóvil.

A causa de la gran diversidad que existe entre modelos, marcas y años de automóviles, no todas las *ECUs* cuentan con el mismo número de parámetros de identificación provocando que el sistema de tablero de instrumentos no logre desplegar toda la información deseada, principalmente se ve reflejado en modelos de vehículos más antiguos dentro del rango en el que opera el escáner.



CONCLUSIONES Y TRABAJO A FUTURO

"Afronta tu camino con coraje, no tengas miedo de las críticas de los demás. Y sobre todo no te dejes paralizar por tus propias críticas"

-Paulo Coelho

Conclusiones.

Trabajo a futuro.





Conclusiones

El objetivo general del proyecto “pantalla *OLED* para información del tablero de instrumentos en vehículos” fue diseñar un sistema que visualice la información del tablero de un automóvil en el parabrisas. Se utilizó una metodología de diseño para abordar dicho proyecto, estudiando los antecedentes necesarios para su comprensión, además de realizar una investigación de sistemas similares, favoreciendo en gran parte la manera de abordar la problemática del sistema.

La etapa de diseño conceptual tomo en cuenta los requerimientos establecidos en un principio y con ayuda de métodos como matrices de conceptos y selección se generó una arquitectura de sistema conteniendo a grandes rasgos los elementos que componen al tablero. De igual forma se designó el modo de trabajo de algunos elementos cómo sería el método de extracción de la información del automóvil por medio de protocolos *OBD-II*, en general se obtuvo un panorama de como sería el principio de funcionamiento del tablero de instrumentos.

Se tomó como base la arquitectura de sistema para desarrollar un modelo funcional que comprobaría la funcionabilidad del mismo, se profundizando en cada uno de los elementos del tablero, como sería el tipo de comunicación que se utiliza dentro del sistema, las interfaces de comunicación inalámbricas (*bluetooth*®) y alámbricas, la lógica del algoritmo desarrollado para la placa *arduino mega*, el procesamiento de datos provenientes de vehículo por medio de los parámetros de identificación, etc.

El desarrollo del modelo funcional, fue costoso la razón es que algunos elementos utilizados se enfocan más en sistemas para realizar prototipos de manera rápida, como es el caso de la placa *arduino mega*. Se podría utilizar sólo el microcontrolador *atmega2560* para un prototipo final, siendo este más barato que la placa, esto mismo ocurre con otros elementos que componen al sistema. Por cuestiones de tiempo se decidió sólo dejar el sistema con un modelo que ejemplifique su funcionamiento.

Para este proyecto no se contó con una pantalla *OLED* con las características necesarias para adaptarse al parabrisas, pero si se trabajó con una pantalla *OLED* a color que no contaba con transparencia ni flexibilidad, resaltando en gran medida la manera de interactuar con ella, como la selección del modo de trabajo, el tipo de comunicación con *arduino* utilizando las librerías específicas, el manejo de la memoria *uSD*, además de comandos necesarios para su control.

El sistema conto de una programación basada en una comunicación *RS232*, la cual se encargaba de enviar, recibir, procesar y nuevamente enviar información de manera constante.



La lectura de datos por la placa arduino es de un carácter en un carácter lo que obligo a guardar toda la trama de respuesta en una cadena, ocasionando algunas veces un desplazamiento en la trama, provocando que al seleccionar el valor para la interpretación del parámetro de identificación, no fuese el correcto. Esto ocurrió en gran parte por la sincronización del sistema, la solución fue buscar un tiempo adecuado para el retardo en el sistema concluyendo en 500[ms] para cada parámetro.

Para le montaje del sistema se implementó un circuito impreso, la cual contiene todo lo necesario para la conexión de la pantalla y el módulo *bluetooth*, exceptuando la alimentación eléctrica, dicha alimentación no exigió mucha demanda de potencia ya que sólo requería de alimentar la placa arduino, la pantalla y el módulo, por esa razón se optó por colocar una batería.

La única manera de comprobar la funcionabilidad del sistema era sometiéndolo a distintas pruebas, estas mismas ayudaron a la detección de errores durante el desarrollo del proyecto. En las pruebas de laboratorio como en las de campo el sistema presento algunos problemas como era la sincronización. De igual forma se comprobó la compatibilidad con distintas marcas y modelos de automóviles siempre y cuando fueran su año de producción mayor al 2000 aunque algunos sobre todo los más cercanos a este año no contaban con algunos parámetros de identificación infiriendo que entre más reciente el modelo del vehículo cuenta con un mayor número de sensores y su *ECU* permite una mayor gama de *PID*'s.

En términos generales el sistema comprobó su funcionabilidad presentando algunos detalles. El objetivo del proyecto era diseñar un sistema que visualizará la información de los vehículos en el parabrisas, no se logró cumplir al 100% este objetivo ya que la pantalla utilizada para el modelo funcional no era la adecuada, aun así se deja una base sólida donde se contempla la adquisición de la información, su procesamiento y la comunicación con la pantalla del tipo *OLED*, la cual en un futuro que se tenga la disponibilidad de pantalla con características de transparencia y flexibilidad, se pueda fabricar una segunda versión que cumpla en absoluto el objetivo general.

La finalidad de desarrollar un proyecto así, es aplicar los conocimientos adquiridos durante la carrera además de presentarse ante nuevos retos demostrando la madurez suficiente para resolverlos. De igual manera se pretende puntualizar sobre el manejo de tecnologías nuevas como las pantallas *OLED* creando aplicaciones como el tablero de instrumentos para vehículos.



Trabajo a futuro

Para trabajar en una segunda versión del tablero de instrumentos se deberán tomar en cuentas varios aspectos que por falta de material o tiempo no se pudieron mejorar durante el desarrollo de esta versión.

- *Las características de la pantalla.*- Es muy importante ya contar con una pantalla que sea del tipo transparente y flexible, la empresa Samsung promete poner en venta las pantallas con estas características que llevan por nombre *YOUM®* a finales del 2012 o principios del 2013.
- *La sincronización del sistema.*- Es de suma importancia verificar esta parte ya que dependiendo el automóvil varía el tiempo que necesita la *ECU* para poder entregar la información, lo que ocasionaba un ajuste del sistema con cada automóvil, lo ideal sería buscar algún tipo de estándar que favoreciera la compatibilidad con un mayor número de vehículos.
- *El montaje del sistema dentro del automóvil.*- Aquí se debe de tomar en cuenta la manera en que se alimentará eléctricamente al sistema preferiblemente del mismo automóvil, una vez contando con la pantalla *OLED* adecuada se debe de considerar la manera de fijar la pantalla al parabrisas además de algún recubrimiento contra fenómenos naturales.

En términos generales estos son los tres puntos principales que se deben de considerar para algún rediseño del sistema de tablero de instrumentos con pantalla *OLED*.



Bibliografía

Libros

- Axelson, J. (1998). *Serial Port Complete*. Madison, Wisconsin, USA.: Lakeview Research.
- Ulrich, K., & Eppinger, S. (2004). *Diseño y Desarrollo de Productos*. Mexico, D.F.: McGraw-Hill.

Tesis

- Cervantes Alonso, I., & Espinosa Solis, S. (2010). *Escaner automotriz de pantalla tactil*. México DF: Instituto Politécnico Nacional ESIME.
- Gonzáles Melis, P. (2008). *Electronica del automovil OBD II*. Barcelona, España: Escuela Tecnica Superior de Ingenieria Industrial de Barcelona.
- Nuñes de la Rosa, E. M., & Barajas, R. (2009). *Diseño de un prototipo de EDR en automoviles con OBDII*. Mexico, DF : Universidad Nacional Autónoma de Mexico, Facultad de Ingeniería.

Artículos

- Carrillo Saucedo, J. C. (2006). *En busca de la pantalla perfecta*. Moreria, México : Universidad de Morelia.
- Murillo, A. (2010). *Tecnología OLED*. Ecuador: Universidad Politecnica Salesiana.

Páginas y manuales electrónicas

- Arduino. (26 de Marzo de 2009). *Esquemático Arduino Mega*. Recuperado el 12 de Julio de 2012, de <http://arduino.cc/es/uploads/Main/arduino-mega-schematic.pdf>
- *Scantools for OBDII vehicles*. (s.f.). Recuperado el 12 de Julio de 2012, de <http://www.obddiagnostics.com/>
- *Sparkfun Electronics*. (s.f.). Recuperado el 12 de Julio de 2012, de <http://www.sparkfun.com/>
- Systems, 4. (7 de Junio de 2012). *Internal 4DGL Functions*. Recuperado el 12 de Julio de 2012, de <http://www.4dsystems.com.au/downloads/Semiconductors/GOLDELOX-GFX2/Docs/GOLDELOX-GFX2-4DGL-Internal-Functions-rev5.pdf>
- *Wikipedia OBD-II PIDs* . (s.f.). Recuperado el 12 de Julio de 2012, de http://en.wikipedia.org/wiki/OBD-II_PIDs



Referencias

- [1] *Organisation Internationale des Constructeurs d'Automobiles*. (s.f.). Recuperado el 12 de Julio de 2012, de <http://www.oica.net/>
- [2] *Mastretta Cars Sitio Oficial*. (s.f.). Recuperado el 12 de Julio de 2012, de <http://mastrettacars.com/mx/>
- [3] *Libera auto*. (s.f.). Recuperado el 12 de Julio de 2012, de <http://www.liberauto.com/articulos/historia/brevisima-historia-del-airbag/art40.aspx>
- [4] Lovins, A. (1998). El hiper-automovil del futuro. Revista *Creces*.
- [5] *Wikipedia Head-Up Display*. (s.f.). Recuperado el 12 de Julio de 2012, de http://en.wikipedia.org/wiki/Head-up_display
- [6] *Peugeot*. (s.f.). Recuperado el 12 de Julio de 2012, de <http://www.peugeot.com/en/products/cars/peugeot-3008.aspx>
- [7] *Auto Blog*. (s.f.). Recuperado el 12 de Julio de 2012, de <http://www.autoblog.com/2010/03/18/video-gm-explains-next-generation-heads-up-display-tech/>
- [8] *Motor Pasión Futuro*. (s.f.). Recuperado el 12 de Julio de 2012, de <http://www.motorpasionfuturo.com/equipamiento-avanzado-gadgets/aida-20-navegador-gps-3d-hud-y-realidad-aumentada-para-audi>
- [9] Alemán Chicaiza, P. M., & Males Cevallos, G. M. (2007). *Estudio de los LEDs orgánicos (OLED), posibles aplicaciones y comparacion con tecnologias disponibles para la presentación de informacion en pantalla planas*. Quito, Ecuador : Escuela Politecnica Nacional.
- [10] *LG*. (s.f.). Recuperado el 12 de Julio de 2012, de <http://www.lg.com/us/oled/index.jsp>
- [11] *Arpem*. (s.f.). Recuperado el 12 de Julio de 2012, de <http://www.arpem.com/coches/coches/mitsubishi/px-miev/px-miev.html>
- [12] *OLED info*. (s.f.). Recuperado el 12 de Julio de 2012, de <http://www.oled-info.com/samsung-flexible-oleds-be-branded-youm>
- [13] Networks, R. (21 de Noviembre de 2009). *Roving Networks Bluetooth*. Recuperado el 12 de Julio de 2012, de <http://www.sparkfun.com/datasheets/Wireless/Bluetooth/rn-bluetooth-um.pdf>

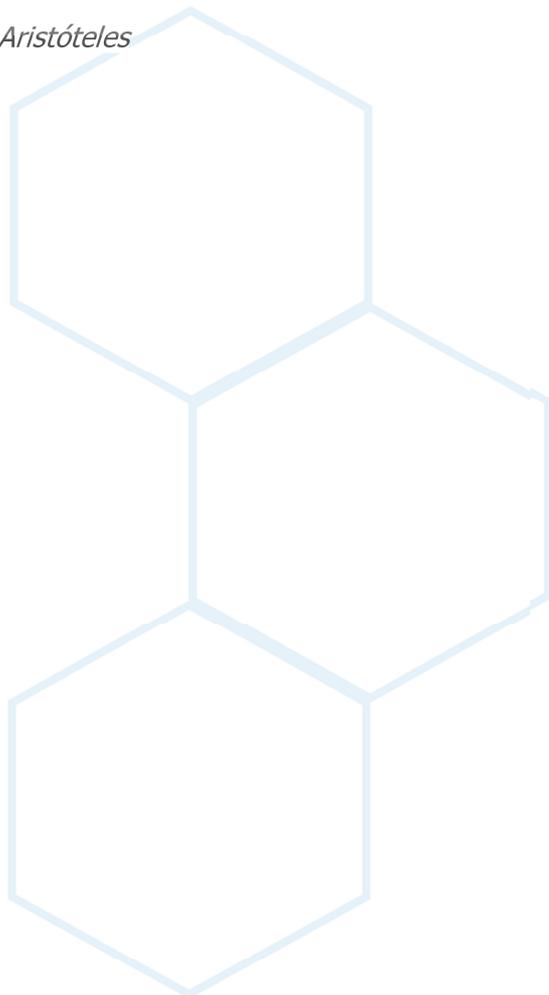


- [14] *Arduino*. (s.f.). Recuperado el 12 de Julio de 2012, de <http://www.arduino.cc/>
- [15] Systems, 4. (16 de Diciembre de 2011). *Platform OLED Display Module Data Sheet*. Recuperado el 12 de Julio de 2012, de [http://www.4dsystems.com.au/downloads/4DGL-Display-Modules/uOLED-160-G1\(GFX\)/Docs/uOLED-160-G1GFX-DS-rev5.pdf](http://www.4dsystems.com.au/downloads/4DGL-Display-Modules/uOLED-160-G1(GFX)/Docs/uOLED-160-G1GFX-DS-rev5.pdf)
- [16] *4D Systems PmmC Loader*. (s.f.). Recuperado el 12 de Julio de 2012, de <http://www.4dsystems.com.au/prod.php?id=46>
- [17] *4D Systems uUSB-MB5*. (s.f.). Recuperado el 12 de Julio de 2012, de <http://www.4dsystems.com.au/prod.php?id=18>
- [18] systems, 4. (17 de Noviembre de 2011). *GOLDELOX-SGC Command Set*. Recuperado el 12 de Julio de 2012, de <http://www.4dsystems.com.au/downloads/Semiconductors/GOLDELOX-SGC/Docs/GOLDELOX-SGC-COMMANDS-SIS-rev6.pdf>
- [19] Gonzalez, O. (s.f.). *Displayshield4d*. Recuperado el 12 de Julio de 2012, de <http://code.google.com/p/displayshield4d/>
- [20] *4D Systems Softwares*. (s.f.). Recuperado el 12 de Julio de 2012, de <http://www.4dsystems.com.au/developers/4dgl-download.php>
- [21] *Cadsoft EAGLE*. (s.f.). Recuperado el 12 de Julio de 2012, de <http://www.cadsoftusa.com/>

Apéndice A

"El ignorante afirma, el sabio duda y reflexiona"

-Aristóteles





Escáner ELM327

El dispositivo *ELM327* es un dispositivo de escaneo automotriz, soporta protocolos *OBD-II*, cuenta con detección de protocolo automático, compatible con todos los vehículos que soporten los protocolos EOBD/OBD2/CAN

Procesador ELM327

El *ELM327* está diseñado para trabajar como puente entre el *OBD (On-Board Diagnostics)* y la interfaces RS232. Adicionalmente cuenta con una detección automática de los protocolos *OBD*, el *ELM327* también soporta comunicaciones de alta velocidad y modo de ahorro de energía.

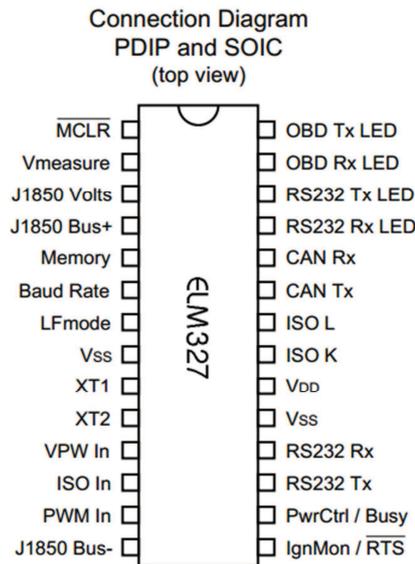


Diagrama A.1 Encapsulado ELM327

Aplicaciones

- Diagnóstico de códigos de falla.
- Herramienta de escaneo automotriz
- Material didáctico

Características

- Búsqueda automática de protocolos
- Configuraciones a través de comandos AT
- Bajo consumo
- Velocidad de 500 kbps
- Interfaz RS232

Diagramas ELM327:

Block Diagram

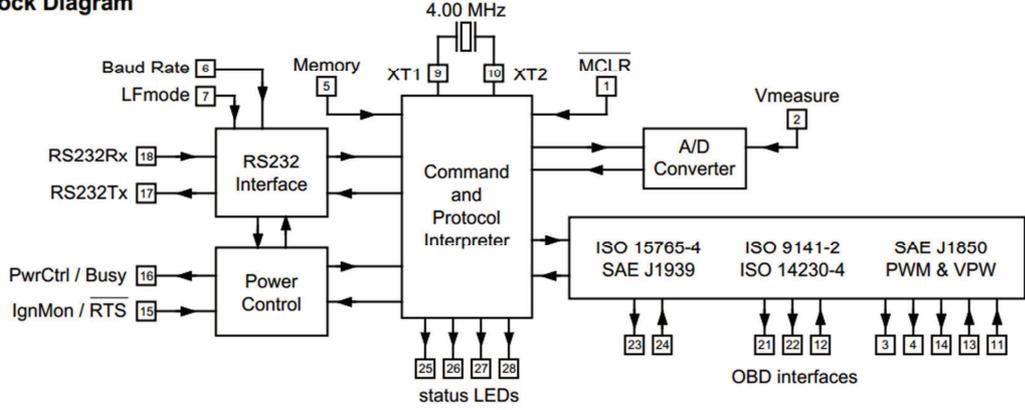


Diagrama A.2 Diagrama de bloques ELM327

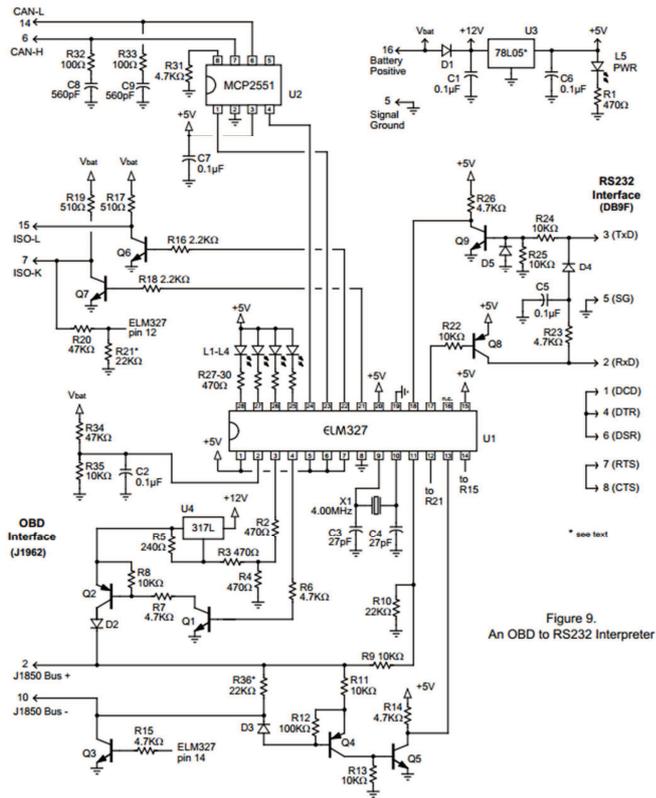


Figure 9. An OBD to RS232 Interpreter

Diagrama A.3 Diagrama interno ELM327



Conector OBD-II

La especificación de *OBD-II* proporciona una interfaz estándar hembra de 16 pines (2x8) conector J1962. SAE J1962 define los *pinout* del conector, como:

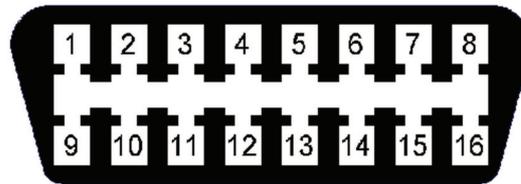


Ilustración A.1 Conector OBD-II

Tabla A.1 Pines Conector OB-II

Terminal	Función
1	En blanco.
2	Bus (+) J1850 VPM y PWM.
3	En blanco.
4	Tierra Chasis.
5	Señal de tierra.
6	CAN alto (J-2284).
7	Línea K ISO 9141-2
8	En blanco
9	En blanco
10	Bus (-) J1850.
11	En blanco
12	En blanco
13	Señal de tierra.
14	CAN bajo (J-2284).
15	Línea L ISO 9141-2
16	Voltaje de la batería.

Modos de medición

El conector de diagnóstico normalizado, deber ser accesible y situarse en la zona del conductor. Los modos de medición son comunes todos los vehículos y permiten desde registrar datos para su verificación, extraer códigos de averías, borrarlos y realizar pruebas dinámicas de actuadores. El software del equipo de diagnóstico se encargará de presentar los datos y facilitar la comunicación. Los modos en que se presentan la información se hayan estandarizado y son siguientes:



Modo 1.- Identificación de parámetro (*PID*), es el acceso a datos en vivo de valores analógicos o digitales de salidas y entradas a la *ECU*. Este modo es también llamado flujo de datos. Aquí es posible ver, por ejemplo, la temperatura de motor o el voltaje generado por una sonda lambda.

Modo 2.- Acceso a cuadro de datos congelados. Esta es una función muy útil del OBD-II porque la *ECU* toma una muestra de todos los valores relacionados con las emisiones, en el momento exacto de ocurrir un fallo. De esta manera, al recuperar estos datos, se pueden conocer las condiciones exactas en las que ocurrió dicho fallo. Sólo existe un cuadro de datos que corresponde al primer fallo detectado.

Modo 3.- Este modo permite extraer de la memoria de la *ECU* todos los códigos de fallo (*DTC - Data Trouble Dode*) almacenados.

Modo 4.- Con este modo se pueden borrar todos los códigos almacenados en la *PCM*, incluyendo los *DTCs* y el cuadro de datos grabados.

Modo 5.- Este modo devuelve los resultados de las pruebas realizadas a los sensores de oxígeno para determinar el funcionamiento de los mismos y la eficiencia del convertidor catalítico.

Modo 6.- Este modo permite obtener los resultados de todas las pruebas de abordó.

Modo 7.- Este modo permite leer de la memoria de la *ECU* todos los *DTCs* pendientes.

Modo 8.- Este modo permite realizar la prueba de actuadores. Con esta función, el mecánico puede activar y desactivar actuadores como bombas de combustible, válvula de ralentí, etc.

Protocolos de comunicación

El sistema *OBD-II* utiliza varios protocolos debido a que existen diversos tipos de computadoras de automóvil. Los protocolos son adoptados por diferentes fabricantes los cuales se ajustan a sus necesidades, aunque también se ha estandarizado, debido a la problemática de compatibilidad. El protocolo a utilizar en un futuro es el llamado *CAN (Controlled Area Network)* el cual establece que un escáner automotriz pueda ser conectado como un nodo dentro de la red del automóvil. En términos generales los más utilizados son:



SAE J1850 PWM

- Creados por *Ford Motors*
- Utiliza la modulación por ancho de pulso a 41.6 kbauds.
- Utiliza dos líneas de comunicación: Bus (+) para comunicación de salida y Bus (-) como comunicación de entrada.
- El bus tiene un valor en voltaje de 0 a 5 volts del modo diferencial.
- Se emplean longitudes de mensaje de 12 bytes tipo CSMA/NDA (*Carrier Sense Multiple Acces/Non Destructible Arbitration*).

SAE J1850 VPM

- Creado por *General Motors*
- Utiliza un bus de ancho de pulso variable con velocidades desde 10.4 hasta 41.6 kbauds.
- Utiliza las mismas líneas de comunicación: Bus (+) y Bus (-).
- Los valores de voltaje son de 2.2 V para un cero lógico y 8 V para un 1 lógico.
- La longitud de mensaje corresponde a la misma de PWM, es decir 12 bytes CSMA/NDA.

ISO9141

- Diseñado por *Chrysler*, vehículos asiáticos y europeos.
- Utiliza una comunicación similar al estándar RS232 con una velocidad de 10.4 kbauds.
- Utiliza las líneas Ky L, aunque al tratarse de una comunicación de dos vías sobre la misma línea se puede suprimir la línea L.
- Los valores de voltaje se ajustan a la batería, inclusive al doble de 24 V.
- De igual forma se restringe a 12 bytes la longitud del mensaje.

ISO 14230 KWP

- Es la evolución del estándar ISO 9141-2.
- La velocidad puede variar de 1.2 a 10.4 kbauds.
- También utiliza las líneas K y L con esta última opcional.
- La longitud del mensaje puede ser hasta de 255 bytes.

ISO 15765 CAN

- Diseñado por la compañía *Bosch*.
- Velocidades de 100 hasta 500 kbauds.
- Crea una pequeña red de comunicaciones entre el auto y la herramienta de diagnóstico.
- Utiliza dos líneas de comunicación: *CAN High* y *CAN Low*.
- Se ha establecido como el protocolo estándar en automóviles fabricados a partir del 2008



Comandos AT

El escáner *ELM327* se interactúa con el a través de comandos AT:

Tabla A.2 Comandos AT

Versión	Comando	Descripción	Grupo
1.0	@1	Display the device description	General
1.3	@2	Display the device identifier	General
1.3	@3 ccccccccccc	Store the device identifier	General
1.0	<CR>	Repeat the last command	General
1.0	1.0 AL	Allow Long (>7 byte) messages	OBD
1.2	AR	Automatic Receive	OBD
1.2	AT0	Adaptive Timing Off	OBD
1.2	AT1	Adaptive Timing Auto1	OBD
1.2	AT2	Adaptive Timing Auto2	OBD
1.0	BD	Perform a Buffer Dump	OBD
1.0	BI	Bypass the Initialization sequence	OBD
1.2	BRD hh	Try Baud rate Divisor hh	General
1.2	BRT hh	Set Baud Rate handshake Timeout	General
1.0	CAF0	CAN Automatic Formatting Off	CAN
1.0	CAF1	CAN Automatic Formatting On	CAN
1.4	CEA	Turn off CAN Extended Addressing	CAN
1.4	CEA hh	Use CAN Extended Address hh	CAN
1.0	CFC1	CAN Flow Control On	CAN
1.4b	CRA	Reset CAN Receive Address filters	CAN
1.4b	CSM0	CAN Silent Mode Off	CAN
1.4b	CSM1	CAN Silent Mode On	CAN
1.0	D	Set all to Defaults	General
1.3	D0	Display of the DLC Off	CAN
1.3	D1	Display of the DLC On	CAN
1.0	DP	Describe the current Protocol	OBD
1.0	DPN	Describe the Protocol by Number	OBD
1.0	E0	Echo Off	General
1.0	E1	Echo On	General
1.1	FC SM h	Flow Control Set the Mode to h	CAN
1.3a	FE	Forget Events	General
1.4	FI	Perform a Fast Initiation	ISO
1.0	H0	Headers Off	OBD
1.0	H1	Headers On	OBD
1.0	I	Print the ID	General
1.0	IB 10	Set the ISO Baud rate to 10400	ISO
1.0	IB 48	Set the ISO Baud rate to 4800	ISO
1.0	IB 96	Set the ISO Baud rate to 9600	ISO
1.2	IFR H	IFR value from Header	J1850
1.2	IFR S	IFR value from Source	J1850
1.2	IFR0	IFRs Off	J1850
1.2	IFR1	IFRs Auto	J1850
1.2	IFR2	IFRs On	J1850
1.4	IGN	Read the IgnMon input level	Other

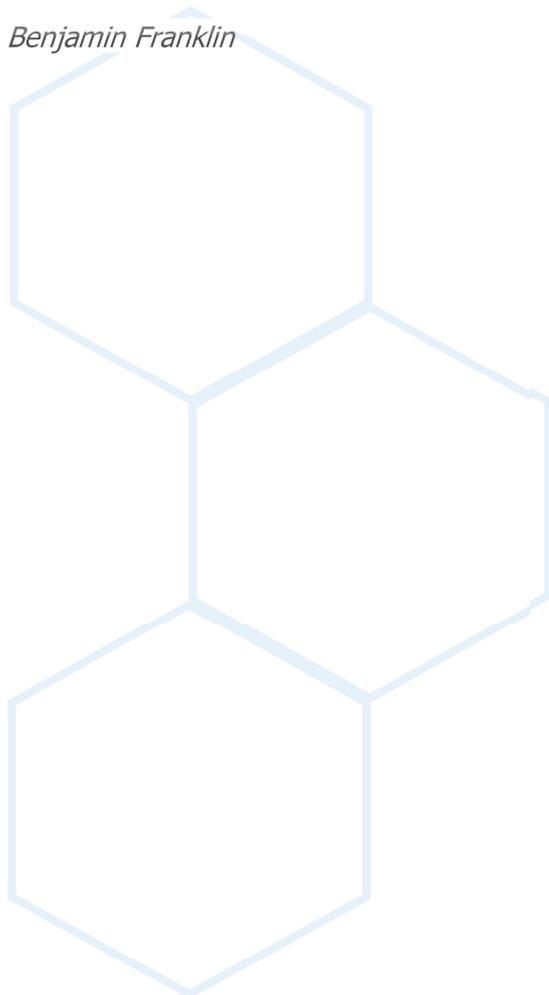


Versión	Comando	Descripción	Grupo
1.2	IIA hh	Set the ISO (slow) Init Address to hh	SO
1.3	JE	Use J1939 Elm data format	J1939
1.4b	JHF0	J1939 Header Formatting Off	J1939
1.4b	JHF1	J1939 Header Formatting On	J1939
1.3	JS	Use J1939 SAE data format	J1939
1.4b	JTM1	Set the J1939 Timer Multiplier to 1x	J1939
1.4b	JTM5	Set the J1939 Timer Multiplier to 5x	J1939
1.3	KW	Display the Key Words	ISO
1.3	KW0	Key Words checking Off	ISO
1.3	KW1	Key Words checking On	ISO
1.0	L0	Linefeeds Off	General
1.0	L1	Linefeeds On	General
1.4	LP	Go to Low Power mode	General
1.0	M0	Memory Off	General
1.0	M1	Memory On	General
1.0	MA	Monitor All	OBD
1.0	MR hh	Monitor for Receiver = hh	OBD
1.0	MT hh	Monitor for Transmitter = hh	OBD
1.0	NL	Normal Length (7 byte) messages	OBD
1.0	PC	Protocol Close	OBD
1.1	PP FF OFF	All Prog Parameters Off	PPs
1.1	PP FF ON	All Prog Parameters On	PPs
1.1	PPS	Print a PP Summary	PPs
1.0	R0	Responses Off	OBD
1.0	R1	Responses On	OBD
1.3	RA hh	Set the Receive Address to hh	OBD
1.4	RD	Read the stored Data	General
1.3	RTR	Send an RTR message	CAN
1.0	RV	Read the Voltage	Volts
1.3	S0	Printing of Spaces Off	OBD
1.3	S1	Printing of Spaces On	OBD
1.4	SD hh	Store Data byte hh	General
1.4	SI	Perform a Slow Initiation	ISO
1.0	SP Ah	Set Protocol to Auto, h and save it	OBD
1.0	SP h	Set Protocol to h and save it	OBD
1.3	SP 00	Set Protocol to Auto and save it	OBD
1.2	SR hh	Set the Receive address to hh	OBD
1.4	SS	Set Standard Search order (J1978)	OBD
1.0	TP Ah	Try Protocol h with Auto search	OBD
1.0	TP h	Try Protocol h	OBD
1.3	V0	Use of Variable DLC Off	CAN
1.3	V1	Use of Variable DLC On	CAN
1.0	Z	Reset all	General

Apéndice B

"Dime y lo olvido, enséñame y lo recuerdo, involúcrame y lo aprendo"

- Benjamin Franklin





OBD-II PID

Son códigos que se utilizan para solicitar datos a un vehículo, que se utilizan como herramienta de diagnóstico. La norma SAE J/1979 define muchos PID, pero los fabricantes también definen *PID*'s muchos más específicos para sus vehículos.

Por lo general, un técnico automotriz usará *PID* con una herramienta de exploración enchufada a través del conector *OBD-II* del vehículo.

1. El técnico ingresa un *PID*.
2. La herramienta de análisis que envía al bus del vehículo (*CAN, VPW, PWM, ISO, KWP*).
3. El dispositivo reconoce el *PID*, genera un bus de datos de respuesta.
4. La herramienta de análisis lee la respuesta, y la muestra para el técnico.

Tabla B.1 Parámetros de identificación OBD-II

Modo (hex)	PID (hex)	Bytes devueltos	Descripción	Unidades	Formula
01	00	4	PIDs Soportados		Bit codificado [A7..D0]=[PID 0x01..PID..0x20]
01	01	4	Número de código de problema e información		Bit codificado.
01	02	8	DTC congelado		
01	03	2	Estado del sistema de combustible		Bit codificado.
01	04	1	Valor calculado de la carga del motor	%	$A*100/255$
01	05	1	Temperatura del anticongelante del motor	°C	$A-40$
01	06	1	Combustible a corto plazo %reducido-	%	$0.7812*(A-128)$
01	07	1	Combustible a largo plazo %reducido-	%	$0.7812*(A-128)$
01	08	1	Combustible a corto plazo %reducido-	%	$0.7812*(A-128)$
01	09	1	Combustible a largo plazo %reducido-	%	$0.7812*(A-128)$
01	0A	1	Presión del combustible	KPa	$A*3$
01	0B	1	Presión del colector de admisión	KPa	A
01	0C	2	RPM del motor	Rpm	$((A*256)+B)/4$
01	0D	1	Velocidad del vehículo	Km/h	A
				relativo al cilindro#1	
01	0E	1	Sincronización de avance		$A/2-64$
01	0F	1	Temperatura del aire de admisión	°C	$A-40$
01	10	2	Flujo de aire del MAF	g/s	$((256*A)+B)/100$
01	11	1	Posición de la válvula reguladora	%	$A*100/255$
01	12	1	Sec. (?) estado del aire		Bit codificado.
01	13	1	Sensores de oxígeno presentes		[A0..A3]=Banco1, Sensores 1- 4[A4..A7]=Banco2.
01	14	2	Banco 1, Sensor 1: Voltaje del sensor de oxígeno, Ajuste a corto plazo del combustible	Volts %	$A*0.005; (B-128)*0.7812$ (si $B=0xFF$, sensor no usado en el ajuste calculado)



01	15	2	Banco 1, Sensor 2: Voltaje del sensor de oxígeno, Ajuste a corto plazo del combustible	Volts %	$A*0.005; (B-128)*0.7812$ (si $B=0xFF$, sensor no usado en el ajuste calculado)
01	16	2	Banco 1, Sensor 3: Voltaje del sensor de oxígeno, Ajuste a corto plazo del combustible	Volts %	$A*0.005; (B-128)*0.7812$ (si $B=0xFF$, sensor no usado en el ajuste calculado)
01	17	2	Banco 1, Sensor 4: Voltaje del sensor de oxígeno, Ajuste a corto plazo del combustible	Volts %	$A*0.005; (B-128)*0.7812$ (si $B=0xFF$, sensor no usado en el ajuste calculado)
01	18	2	Banco 2, Sensor 1: Voltaje del sensor de oxígeno, Ajuste a corto plazo del combustible	Volts %	$A*0.005; (B-128)*0.7812$ (si $B=0xFF$, sensor no usado en el ajuste calculado)
01	19	2	Banco 2, Sensor 2: Voltaje del sensor de oxígeno, Ajuste a corto plazo del combustible	Volts %	$A*0.005; (B-128)*0.7812$ (si $B=0xFF$, sensor no usado en el ajuste calculado)
01	1A	2	Banco 2, Sensor 3: Voltaje del sensor de oxígeno, Ajuste a corto plazo del combustible	Volts %	$A*0.005; (B-128)*0.7812$ (si $B=0xFF$, sensor no usado en el ajuste calculado)
01	1B	2	Banco 2, Sensor 4: Voltaje del sensor de oxígeno, Ajuste a corto plazo del combustible	Volts %	$A*0.005; (B-128)*0.7812$ (si $B=0xFF$, sensor no usado en el ajuste calculado)
01	1C	1	Los estándares OBD del vehiculo se		Bit codificado,
01	1D	1	Sensores de oxígeno presentes		Similar al PID13, pero [A0..A7]=[B1S1, B1S2, B2S1, B2S2, B3S1, B3S2, B4S1, B4S2]
01	1E	1	Estado de la entrada auxiliar		A0==toma de alimentación apagada (PTO) estado (1==activo) [A1..A7]no usada
01	1F	2	Tiempo de ejecución desde el arranque del motor	Seg	$(A*256)+B$
01	20	4	PIDs soportados 21-40		Bit codificado [A7..D0]=[PID ox21..PID0x40]
01	21	2	Distancia recorrida con la luz indicadora de mal funcionamiento (MIL) encendida	Km	$(A*256)+B$
01	22	2	Carril de presión de combustible (en relación)	KPa	$((A*256)+B)*0.079$
01	23	2	Carril de presión de combustible (diesel)	KPa	$((A*256)+B)*10$
01	24	4	O2S1_WR_lambda(1) Relación de equivalencia de voltaje	N/A V	$((A*256)+B)*0.0000305$ $((C*256)+D)*0.000122$
01	25	4	O2S2_WR_lambda(1) Relación de equivalencia de voltaje	N/A V	$((A*256)+B)*0.0000305$ $((C*256)+D)*0.000122$
01	26	4	O2S3_WR_lambda(1) Relación de equivalencia de voltaje	N/A V	$((A*256)+B)*0.0000305$ $((C*256)+D)*0.000122$
01	27	4	O2S4_WR_lambda(1) Relación de equivalencia de voltaje	N/A V	$((A*256)+B)*0.0000305$ $((C*256)+D)*0.000122$
01	28	4	O2S5_WR_lambda(1) Relación de equivalencia de voltaje	N/A V	$((A*256)+B)*0.0000305$ $((C*256)+D)*0.000122$
01	29	4	O2S6_WR_lambda(1) Relación de equivalencia de voltaje	N/A V	$((A*256)+B)*0.0000305$ $((C*256)+D)*0.000122$



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO



01	2A	4	O2S7_WR_lambda(1) Relación de equivalencia de voltaje	N/A V	$((A*256)+B)*0.0000305$ $((C*256)+D)*0.000122$
01	2B	4	O2S8_WR_lambda(1) Relación de equivalencia de voltaje	N/A V	$((A*256)+B)*0.0000305$ $((C*256)+D)*0.000122$
01	2C	1	Mando EGR	%	$100*A/255$
01	2D	1	Error EGR	%	$A*.78125-100$
01	2E	1	Mando de purgación evaporada	%	$100*A/255$
01	2F	1	Nivel de entrada de combustible	%	$100*A/255$
01	30	1	# de calentamientos desde el borrado de códigos	N/A	A
01	31	2	Distancia recorrida desde el borrado de códigos	Km	$(A*256+B)$
01	32	2	Presión del sistema de evaporación de gases	Pa	$((A*256)+B)/4-8,192$
01	33	1	Presión barométrica	KPa	A
01	34	4	O2S1_WR_lambda(1) Relación de equivalencia de corriente	N/A mA	$((A*256)+B)*0.0000305$ $((C*256)+D)*0.00391-128$
01	35	4	O2S2_WR_lambda(1) Relación de equivalencia de corriente	N/A mA	$((A*256)+B)*0.0000305$ $((C*256)+D)*0.00391-128$
01	36	4	O2S3_WR_lambda(1) Relación de equivalencia de corriente	N/A mA	$((A*256)+B)*0.0000305$ $((C*256)+D)*0.00391-128$
01	37	4	O2S4_WR_lambda(1) Relación de equivalencia de corriente	N/A mA	$((A*256)+B)*0.0000305$ $((C*256)+D)*0.00391-128$
01	38	4	O2S5_WR_lambda(1) Relación de equivalencia de corriente	N/A mA	$((A*256)+B)*0.0000305$ $((C*256)+D)*0.00391-128$
01	39	4	O2S6_WR_lambda(1) Relación de equivalencia de corriente	N/A mA	$((A*256)+B)*0.0000305$ $((C*256)+D)*0.00391-128$
01	3A	4	O2S7_WR_lambda(1) Relación de equivalencia de corriente	N/A mA	$((A*256)+B)*0.0000305$ $((C*256)+D)*0.00391-128$
01	3B	4	O2S8_WR_lambda(1) Relación de equivalencia de Corriente	N/A mA	$((A*256)+B)*0.0000305$ $((C*256)+D)*0.00391-128$
01	3C	2	Temperatura del catalizador Banco 1, sensor 1	°C	$((A*256)+B)/10-40$
01	3D	2	Temperatura del catalizador Banco 2, sensor 1	°C	$((A*256)+B)/10-40$
01	3E	2	Temperatura del catalizador Banco 1, sensor 2	°C	$((A*256)+B)/10-40$
01	3F	2	Temperatura del catalizador Banco 2, sensor 2	°C	$((A*256)+B)/10-40$
01	40	4	PIDs soportados 41-60 (?)		Bit codificado $[A7..D0]=[PID\ 0x41..PID$

Apéndice B: PARÁMETROS DE IDENTIFICACIÓN OBD-II



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

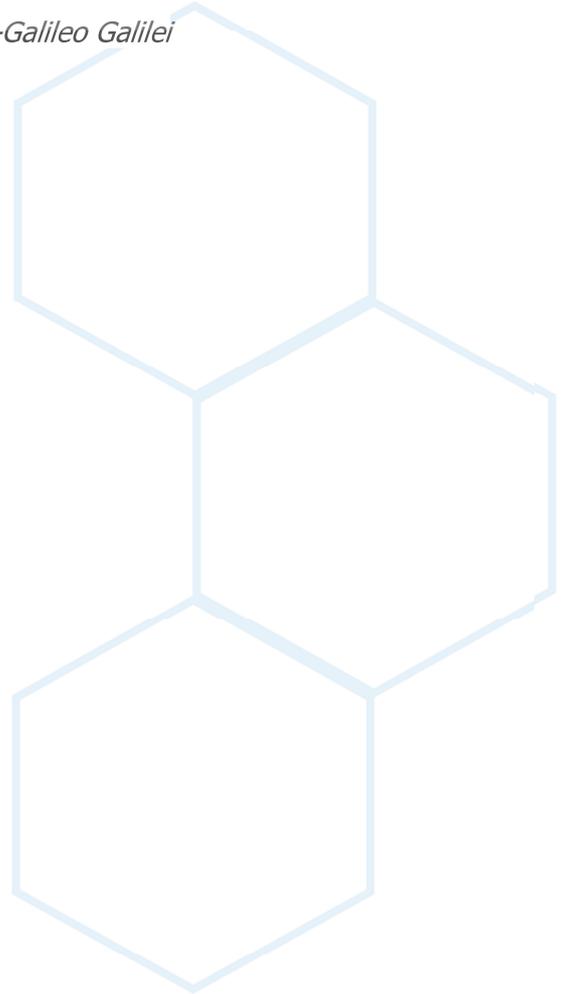


01	42	2	Módulo de control de voltaje	V	$((A*256)+B)/1000$
01	43	2	Valor absoluto de carga	%	$((A*256)+B)*100/255$
01	44	2	Comando de relación de equivalencia	N/A	$((A*256)+B)*0.0000305$
01	45	1	Posición relativa de la válvula reguladora	%	$A*100/255$
01	46	1	Temperatura del aire ambiente	°C	A-40
01	47	1	Posición absoluta B de la válvula reguladora	%	$A*100/255$
01	48	1	Posición absoluta C de la válvula reguladora	%	$A*100/255$
01	49	1	Posición D del pedal acelerador	%	$A*100/255$
01	4A	1	Posición E del pedal acelerador	%	$A*100/255$
01	4B	1	Posición F del pedal acelerador	%	$A*100/255$
01	4C	1	Comando del actuador de la válvula reguladora	%	$A*100/255$
01	4D	2	Tiempo de ejecución con la luz indicadora de mal funcionamiento (MIL) encendida	minutos	$(A*256)+B$
01	4E	2	Problema de tiempo desde el borrado de códigos	minutos	$(A*256)+B$
01	C4	?	?	?	Motor inactivo, petición B5 Motor detenido, petición B6
02	02	2	Código de problema de congelamiento		BCD codificados.
03	N/A	N*6	Solicitud de códigos de fallas		3 códigos por mensaje, codificado BCD
04	N/A	0	Borrar los códigos de falla/Luz indicadora de mal funcionamiento (MIL)/compruebe la luz del motor		Borrar todos los códigos de falla almacenados y apagar la luz indicadora de mal funcionamiento
09	02	5x5	Número de identificación del vehículo		Devuelve 5 líneas, A es la línea de solicitud de bandera, B-E dígitos VIN cifrados en ASCII.

Apéndice C

"Las matemáticas son el alfabeto con el cual Dios ha escrito el Universo"

-Galileo Galilei





Algoritmo de programación desarrollado en el software de arduino 1.0 para la etapa de procesamiento de información y visualización de la información en la pantalla.

```
#include <displayshield4d.h> // Librería para pantalla OLED

char cadena[15]; // Cadena de 16 caracteres
char X; // Variable de tipo char
char Y; // Variable de tipo char
int A; // Variable de tipo entera
int B; // Variable de tipo entera
int C; // Variable de tipo entera
int D; // Variable de tipo entera
int E; // Variable de tipo entera
int F; // Variable de tipo entera
int G; // Variable de tipo entera
int H; // Variable de tipo entera
int I; // Variable de tipo entera
int VEL; // Variable de tipo entera para la velocidad del automóvil
float RPM; // Variable de tipo flotante para las revoluciones del motor
int FUEL; // Variable de tipo entera para el combustible
int i; // Variable de tipo entera
int h; // Variable de tipo entera
int l; // Variable de tipo entera
int m; // Variable de tipo entera
int Cent; // Variable de tipo entera de centenas
int Dec; // Variable de tipo entera de decenas
int Decin; // Variable de tipo entera del residuo de decenas
int Uni; // Variable de tipo entera de unidades
int num[]={0x00,0x10,0x20,0x30,0x40,0x50,0x60,0x70,0x80,0x90}; // Arreglo de variables enteras

DisplayShield4d oled; // Librería OLED
void setup(){ // Función setup

  Serial1.begin(9600); // Iniciación del Puerto serial 1 a 9600 baudios
  Serial.begin(9600); // Iniciación del Puerto serial 0 a 9600 baudios
  Serial1.flush(); // Limpieza del Puerto serial 1
  Serial.flush(); // Limpieza del Puerto serial 0
  oled.Init(); // Inicio de comunicación pantalla OLED
  oled.Clear(); // Limpieza de la pantalla OLED
} // Fin de la función setup

void loop() { // Función loop
  Serial1.println("AT AR"); // Envío de comando "AT AR" por Puerto 1
  delay(500); // Retardo
  if (Serial1.available() > 0){ // Verificar si el Puerto 1 contiene datos
    /* Lectura del Puerto 1
    cadena[0] = Serial1.read();
    cadena[1] = Serial1.read();
    cadena[2] = Serial1.read();
    cadena[3] = Serial1.read();
    cadena[4] = Serial1.read();
    cadena[5] = Serial1.read();
    cadena[6] = Serial1.read();
    cadena[7] = Serial1.read();
    cadena[8] = Serial1.read();
    cadena[9] = Serial1.read();
    cadena[10] = Serial1.read();
    cadena[11] = Serial1.read();
    cadena[12] = Serial1.read();
    cadena[13] = Serial1.read();
    cadena[14] = Serial1.read();
    cadena[15] = Serial1.read();
    Lectura del Puerto 1*/
```



```
char X = char(cadena[6]);           // Cambio de variable
char Y = char(cadena[7]);           // Cambio de variable
oled.drawstringblock(20,20,0,oled.RGB(255,255,255),2,2,"BIENVENIDO"); //Comando de texto pantalla OLED
delay(500);                          // Retardo
oled.Clear();                          // Limpieza de la pantalla OLED
}
Serial1.flush();                       // Limpieza del Puerto serial 1

Serial1.println("AT SP 00");           // Envío de comando "AT SP 00" por Puerto 1
delay(500);                            // Retardo
if (Serial1.available() > 0) {         // Verificar si el Puerto 1 contiene datos
  cadena[0] = Serial1.read();          /* Lectura del Puerto 1
  cadena[1] = Serial1.read();
  cadena[2] = Serial1.read();
  cadena[3] = Serial1.read();
  cadena[4] = Serial1.read();
  cadena[5] = Serial1.read();
  cadena[6] = Serial1.read();
  cadena[7] = Serial1.read();
  cadena[8] = Serial1.read();
  cadena[9] = Serial1.read();
  cadena[10] = Serial1.read();
  cadena[11] = Serial1.read();
  cadena[12] = Serial1.read();
  cadena[13] = Serial1.read();
  cadena[14] = Serial1.read();
  cadena[15] = Serial1.read();        Lectura del Puerto 1 */

  char X = char(cadena[9]);           // Cambio de variable
  char Y = char(cadena[10]);          // Cambio de variable
  oled.drawstringblock(20,20,0,oled.RGB(255,255,255),2,2,"INICIANDO"); //Comando de texto pantalla OLED
  delay(500);                          // Retardo
  oled.Clear();                          // Limpieza de la pantalla OLED
}

while(VEL!='-1'){                     // Inicialización del ciclo while

Serial1.flush();                       // Limpieza del Puerto serial 1
Serial1.println("010D");               // Envío de comando "010D" por Puerto 1
delay(250);                            // Retardo
if (Serial1.available() > 0) {         // Verificar si el Puerto 1 contiene datos
  cadena[0] = Serial1.read();          /* Lectura del Puerto 1
  cadena[1] = Serial1.read();
  cadena[2] = Serial1.read();
  cadena[3] = Serial1.read();
  cadena[4] = Serial1.read();
  cadena[5] = Serial1.read();
  cadena[6] = Serial1.read();
  cadena[7] = Serial1.read();
  cadena[8] = Serial1.read();
  cadena[9] = Serial1.read();
  cadena[10] = Serial1.read();
  cadena[11] = Serial1.read();
  cadena[12] = Serial1.read();
  cadena[13] = Serial1.read();
  cadena[14] = Serial1.read();
  cadena[15] = Serial1.read();        Lectura del Puerto 1 */
  delay(250);                          // Retardo
```



```
int A = (cadena[10]);           // Cambio de variable
int B = (cadena[11]);           // Cambio de variable
if(A>'9'){ A = A-55;}           /* Rutina de cambio a ACSII
else{A = A-48;}
if(B>'9'){B = B-55;}
else{B = B-48;}                */
int VEL = ((A*16)+B);           // Ecuación de parámetro de identificación Velocidad
Cent = (VEL%100);               // Selección del valor de las centenas
Decin =(VEL/10);                // Residuo del valor de las decenas
Dec =(Decin%10);                // Selección del valor de las decenas
Uni = (VEL%10);                 // Selección de valor de las unidades

delay(500);                       // Retardo
oled.ScrCopyDis(0x78,0x00,0x29,0x32,0x0F,0x78,0x21); // Comando despliegue de imagen en pantalla "Km/h"
oled.ScrCopyDis(0x00,0x32,0x4E,0x4E,0x0F,0x78,0x2A); // Comando despliegue de imagen en pantalla "tacómetro"
oled.ScrCopyDis(0x4E,0x32,0x14,0x4E,0x0F,0x78,0x49); // Comando despliegue de imagen en pantalla "RPM"
oled.ScrCopyDis(0x8A,0x32,0x16,0x4E,0x0F,0x78,0x42); // Comando despliegue de imagen en pantalla "FUEL"
oled.ScrCopyDis(0x00,0x00,0x28,0x32,0x0F,0x7A,num[Cent]); // Comando despliegue de imagen en pantalla centenas
oled.ScrCopyDis(0x28,0x00,0x28,0x32,0x0F,0x7A,num[Dec]); // Comando despliegue de imagen en pantalla decenas
oled.ScrCopyDis(0x50,0x00,0x28,0x32,0x0F,0x7A,num[Uni]); // Comando despliegue de imagen en pantalla unidades
}

Serial1.flush();                 // Limpieza del Puerto serial 1
Serial1.println("010C");         // Envío de comando "010C" por Puerto 1
delay(250);                       // Retardo
if (Serial1.available() > 0) {   // Verificar si el Puerto 1 contiene datos
  cadena[0] = Serial1.read();     /* Lectura del Puerto 1
  cadena[1] = Serial1.read();
  cadena[2] = Serial1.read();
  cadena[3] = Serial1.read();
  cadena[4] = Serial1.read();
  cadena[5] = Serial1.read();
  cadena[6] = Serial1.read();
  cadena[7] = Serial1.read();
  cadena[8] = Serial1.read();
  cadena[9] = Serial1.read();
  cadena[10] = Serial1.read();
  cadena[11] = Serial1.read();
  cadena[12] = Serial1.read();
  cadena[13] = Serial1.read();
  cadena[14] = Serial1.read();
  cadena[15] = Serial1.read();   Lectura del Puerto 1*/
  delay(250);                   // Retardo
  int C = (cadena[10]);          // Cambio de variable
  int D = (cadena[11]);          // Cambio de variable
  int E = (cadena[13]);          // Cambio de variable
  int F = (cadena[14]);          // Cambio de variable
  if(C>'9'){C = C-55;}           /* Rutina de cambio a ACSII
  else{C = C-48;}
  if(D>'9'){D = D-55;}
  else{D = D-48;}
  if(E>'9'){E = E-55;}
  else{E = E-48;}
  if(F>'9'){F = F-55;}
  else{F = F-48;}                */

float RPM=((((C*16)+D)*256)+((E*16)+F))*0.25)/1000; // Ecuación de parámetro de identificación RPM
```



```

if(RPM >= 0 && RPM < 0.40) {h=6; l=105;}
else if(RPM >= 0.40 && RPM < 0.70) {h=3; l=101;}
else if(RPM >= 0.70 && RPM < 1.00) {h=2; l=96;}
else if(RPM >= 1.00 && RPM < 1.25) {h=1; l=92;}
else if(RPM >= 1.25 && RPM < 1.50) {h=1; l=87;}
else if(RPM >= 1.50 && RPM < 1.75) {h=1; l=82;}
else if(RPM >= 1.75 && RPM < 2.00) {h=2; l=77;}
else if(RPM >= 2.00 && RPM < 2.25) {h=4; l=72;}
else if(RPM >= 2.25 && RPM < 2.50) {h=7; l=67;}
else if(RPM >= 2.50 && RPM < 2.75) {h=10; l=62;}
else if(RPM >= 2.75 && RPM < 3.00) {h=13; l=60;}
else if(RPM >= 3.00 && RPM < 3.25) {h=17; l=57;}
else if(RPM >= 3.25 && RPM < 3.50) {h=21; l=54;}
else if(RPM >= 3.50 && RPM < 3.75) {h=26; l=52;}
else if(RPM >= 3.75 && RPM < 4.00) {h=31; l=51;}
else if(RPM >= 4.00 && RPM < 4.25) {h=37; l=50;}
else if(RPM >= 4.25 && RPM < 4.50) {h=43; l=50;}
else if(RPM >= 4.50 && RPM < 4.75) {h=49; l=51;}
else if(RPM >= 4.75 && RPM < 5.00) {h=53; l=53;}
else if(RPM >= 5.00 && RPM < 5.25) {h=57; l=55;}
else if(RPM >= 5.25 && RPM < 5.50) {h=61; l=58;}
else if(RPM >= 5.50 && RPM < 5.75) {h=66; l=61;}
else if(RPM >= 5.75 && RPM < 6.00) {h=69; l=66;}
else if(RPM >= 6.00 && RPM < 6.25) {h=71; l=70;}
else if(RPM >= 6.25 && RPM < 6.50) {h=73; l=75;}
else if(RPM >= 6.50 && RPM < 6.75) {h=75; l=79;}
else if(RPM >= 6.75 && RPM < 7.00) {h=75; l=85;}
else if(RPM >= 7.00 && RPM < 7.25) {h=75; l=90;}
else if(RPM >= 7.25 && RPM < 7.50) {h=74; l=95;}
else if(RPM >= 7.50 && RPM < 7.75) {h=72; l=100;}
else if(RPM >= 7.75 && RPM < 8.00) {h=70; l=105;}
else if(RPM >= 8.00 && RPM < 8.25) {h=68; l=109;}
else if(RPM >= 8.25 && RPM < 8.50) {h=65; l=112;}
else if(RPM >= 8.50 && RPM < 8.75) {h=62; l=117;}
else if(RPM >= 8.75 && RPM < 9.00) {h=52; l=122;}
else if(RPM >= 9.00){h=6; l=105;}
else{h=5; l=105;}

/* Selección de imagen RPM

delay(500); // Retardo
oled.circle(39, 89, 4, OLED_SOLID, oled.RGB(255, 0, 0)); // Comando dibujar un circulo en la pantalla
oled.line(39, 89, h, l, oled.RGB(255,0,0)); // Comando dibujar una línea en la pantalla
}

Serial1.flush(); // Limpieza del Puerto serial 1
Serial1.println("012F"); // Envió de comando "010C" por Puerto 1
delay(250); // Retardo
if (Serial1.available() > 0 ){ // Verificar si el Puerto 1 contiene datos
  cadena[0] = Serial1.read(); /* Lectura del Puerto 1
  cadena[1] = Serial1.read();
  cadena[2] = Serial1.read();
  cadena[3] = Serial1.read();
  cadena[4] = Serial1.read();
  cadena[5] = Serial1.read();
  cadena[6] = Serial1.read();
  cadena[7] = Serial1.read();
  cadena[8] = Serial1.read();
  cadena[9] = Serial1.read();
  cadena[10] = Serial1.read();
  cadena[11] = Serial1.read();
  cadena[12] = Serial1.read();
  cadena[13] = Serial1.read();
  cadena[14] = Serial1.read();
  cadena[15] = Serial1.read(); Lectura del Puerto 1*/

```



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO



```
delay(250); // Retardo
int G = (cadena[10]); // Cambio de variable
int H = (cadena[11]); // Cambio de variable
if(G>'9'){G = G-55;} /* Rutina de cambio a ACSII
else{G = G-48;}
if(B>'9'){H = H-55;}
else{H = H-48;} */

int FUEL = (100*((G*16)+H))/255; // Ecuación de parámetro de identificación Combustible

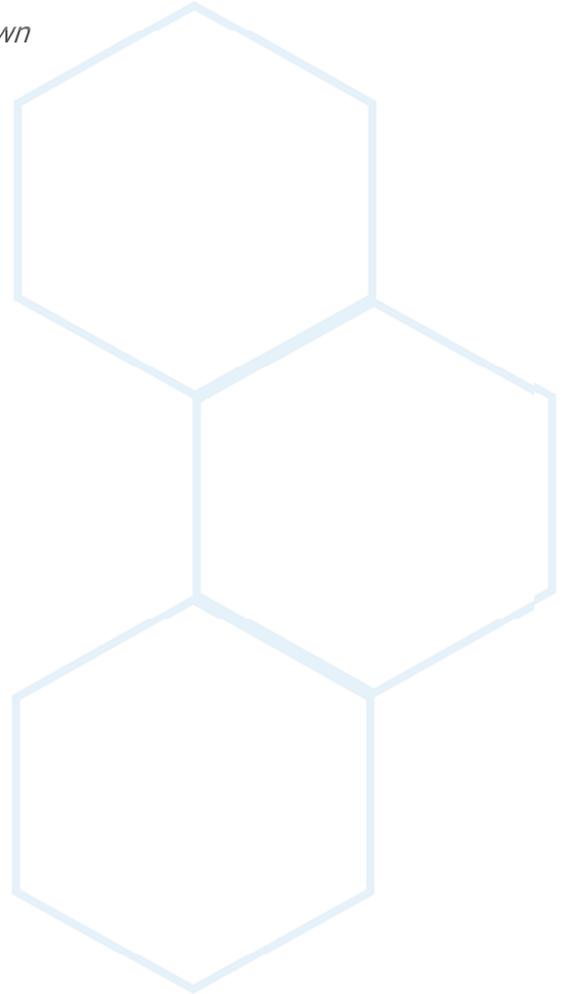
if(FUEL >= 0 && FUEL < 20){m=0x00;} /* Selección de la imagen combustible
else if(FUEL >= 20 && FUEL < 40){m=0x10;}
else if(FUEL >= 40 && FUEL < 60){m=0x20;}
else if(FUEL >= 60 && FUEL < 80){m=0x30;}
else if(FUEL >= 80 && FUEL <= 100){m=0x40;}
else {m=0x00;} */

delay(500); // Retardo
oled.ScrCopyDis(0x62,0x32,0x2A,0x4E,0x0F,0x7C,m); // Comando de despliegue de imagen combustible
}
} // Fin del ciclo while
} // Fin de la función loop
```

Apéndice D

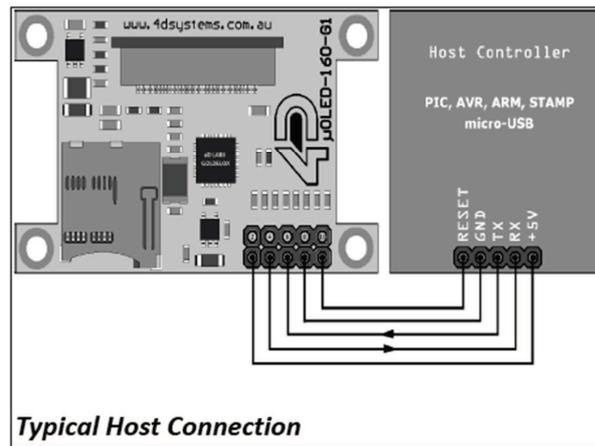
"Pregúntate si lo que estás haciendo hoy te acerca al lugar en el que quieres estar mañana"

-J. Brown



La pantalla *OLED* de la marca *4D systems* cuando trabaja en modo *SGC* es controlado a través de un microcontrolador con el envío de un conjunto de comandos. La interfaz de comandos entre el *GOLDELOX-SGC* y el microcontrolador es a través de una interfaz en serie. Todos los comandos devuelven una respuesta, ya sea un reconocimiento o de datos.

La conexión entre la pantalla *OLED* y el maestro es de la siguiente manera:



El conjunto de comandos se agrupan en las siguientes secciones:

- **Comandos generales.**
 - AutoBaud – **55hex**
 - Version-Device Info Request – **56hex**
 - Replace Background Colour – **42hex**
 - Clear Screen – **45hex**
 - Display Control Functions – **59hex**
 - Sleep– **5Ahex**
 - Switch-Buttons-Joystick Status - **4Ahex**
 - Switch-Buttons-Joystick Wait for Status - **6Ahex**
 - Sound – **4Ehex**
 - Tune – **6Ehex**
- **Comandos gráficos.**
 - Add User Bitmap Character – **41hex**
 - Draw Circle – **43hex**
 - Draw User Bitmap Character – **44hex**
 - Draw Triangle – **47hex**
 - Draw Image-Icon – **49hex**
 - Set Background colour – **4Bhex**
 - Draw Line – **4Chex**
 - Draw Pixel – **50hex**
 - Read Pixel – **52hex**



- Screen Copy-Paste – **63hex**
- Draw Polygon – **67hex**
- Replace colour – **6Bhex**
- Set Pen Size – **70hex**
- Draw Rectangle – **72hex**
- **Comandos de texto.**
 - Set Font – **46hex**
 - Set Transparent-Opaque Text – **4Fhex**
 - Draw “String” of ASCII Text (graphics format) – **53hex**
 - Draw ASCII Character (text format) – **54hex**
 - Draw Text Button – **62hex**
 - Draw “String” of ASCII Text (text format) – **73hex**
 - Draw ASCII Character (graphics format) – **74hex**
- **Comandos de memory card (uSD)**
 - Set Address Pointer of Memory Card – **@41hex**
 - Screen Copy-Save to Memory Card - **@43hex**
 - Display Image-Icon from Memory Card - **@49hex**
 - Display Object from Memory Card - **@4Fhex**
 - Run Script (4DSL) Program from Memory Card - **@50hex**
 - Read Sector Block Data from Memory Card - **@52hex**
 - Display Video-Animation Clip from Memory Card - **@56hex**
 - Write Sector Block Data to Memory Card - **@57hex**
 - Initialise Memory Card - **@69hex**
 - Read Byte Data from Memory Card - **@72hex**
 - Write Byte Data to Memory Card - **@77hex**

A continuación se mostrarán la descripción más detallada de algunos comandos:

Comandos generales

AutoBaud – 55hex

Command	cmd	
4DSL Cmd	AutoBaud	
	cmd	55(hex) or U(ascii) : Command header byte
Response	acknowledge	
	acknowledge	06(hex) : ACK byte
Description	This must be the very first command sent to the GOLDELOX-SGC after power-up or reset. This will enable the device to lock on to the host baud rate.	
Serial Example	Command Data: 55hex Send Autobaud command.	



Clear Screen – 45hex

Command	cmd	
4DSL Cmd	Clear	
	cmd	45(hex) or E(ascii) : Command header byte
Response	acknowledge	
	acknowledge	06(hex) : ACK byte if successful 15(hex) : NAK byte if unsuccessful
Description	This command clears the entire screen using the current background colour	
Serial Example	Command Data: 45hex (Clear the screen).	

Display Control Functions – 59hex

Command	cmd, mode, value	
4DSL Cmd	Control(mode, value)	
	cmd	59(hex) or Y(ascii) : Command header byte
	mode	00hex : NA 01hex : Display ON/OFF DISPLAY OFF : when value = 00hex DISPLAY ON : when value = 01hex 02hex : Contrast Adjust CONTRAST RANGE : when value = 00hex to 0Fhex 03hex : Display PowerUp-Shutdown (low power mode) DISPLAY SHUTDOWN : when value = 00hex DISPLAY POWERUP : when value = 01hex
	value	See mode description above.
Response	acknowledge	
	acknowledge	06(hex) : ACK byte if successful 15(hex) : NAK byte if unsuccessful
Description	This command changes some of the display settings such as contrast and low power mode.	
Serial Example	Command Data: 59hex, 01hex, 00hex Turn the Display off.	



Comandos Gráficos

Draw Circle – 43hex

Command	cmd, x, y, radius, colour(msb:lsb)	
4DSL Cmd	Circle(x, y, radius, colour)	
	cmd	43(hex) or C(ascii) : Command header byte
	x	Horizontal position of the circle centre.
	y	Vertical position of the circle centre.
	radius	Radius of the circle.
	colour	2 bytes define the circle colour.
Response	acknowledge	
	acknowledge	06(hex) : ACK byte if successful 15(hex) : NAK byte if unsuccessful
Description	This command will draw a coloured circle centred at (x, y) with a radius determined by the value set in the ' radius ' byte. The circle can be either solid or wire frame (empty) depending on the value of the Pen Size (see Set Pen Size command). when Pen Size = 0 : circle is solid when Pen Size = 1 : circle is wire frame	
Serial Example	Command Data: 43hex, 3Fhex, 3Fhex, 22hex, 00hex, 1Fhex Draws a RED circle (001Fhex) centred at x = 63dec (3Fhex) and y = 63de (3Fhex) with a radius of 34dec (22hex) .	



Draw Triangle – **47hex**

Command	cmd, x1, y1, x2, y2, x3, y3, colour(msb:lsb)	
4DSL Cmd	Triangle(x1, y1, x2, y2, x3, y3, colour)	
	cmd	47(hex) or G(ascii) : Command header byte
	x1, y1, x2, y2,	3 vertices of the triangle. These must be specified in an anti-clockwise
	colour	2 bytes (big endian) triangle colour value.
Response	acknowledge	
	acknowledge	06(hex) : ACK byte if successful 15(hex) : NAK byte if unsuccessful
Description	<p>This command draws a Solid/Wire-Frame triangle. The vertices must be specified in an anti-clock wise manner, i.e.</p> <p style="text-align: center;">x2 < x1 : x3 > x2 : y2 > y1 : y3 > y1</p> <p>A solid or a wire frame triangle is determined by the value of the Pen Size setting.</p> <p>when Pen Size = 0 : triangle is solid when Pen Size = 1 : triangle is wire frame</p>	
Serial Example	Command Data: 47hex, 25hex, 10hex, 05hex, 30hex, 35hex, 35hex, F8hex, 00hex (Display a RED triangle).	

Draw Line – **4Chex**

Command	cmd, x1, y1, x2, y2, colour(msb:lsb)	
4DSL Cmd	Line(x1, y1, x2, y2, colour)	
	cmd	4C(hex) or L(ascii) : Command header byte
	x1	Top left horizontal start position of line.
	y1	Top left vertical start position of line.
	x2	Bottom right horizontal end position of line.
	y2	Bottom right vertical end position of line.
	colour	2 bytes define the Line colour.
Response	acknowledge	
	acknowledge	06(hex) : ACK byte if successful 15(hex) : NAK byte if unsuccessful
Description	This command will draw a coloured line from point (x1, y1) to point (x2, y2) on the screen.	
Serial Example	Command Data: 4Chex, 00hex, 00hex, 7Fhex, 7Fhex, Ffhex, Ffhex Draws a WHITE line (FFFFhex) from (x1 = 00hex, y1 = 00hex) to (x2 = 7Fhex, y2 = 7Fhex).	



Draw Polygon – **67hex**

Command	cmd, vertices, x1, y1, .. , xn, yn, colour(msb:lsb)	
4DSL Cmd	Polygon(vertices, x1, y1, .. , xn, yn, colour)	
	cmd	67(hex) or g(ascii) : Command header byte
	vertices	Number of vertices from 3 to 7. This byte specifies the number of vertices of the polygon.
	x1,y1,..xn, yn	Vertices of the Polygon. These can be specified in any fashion. 1 Byte each.
	colour	2 bytes triangle colour value.
Response	acknowledge	
	acknowledge	06(hex) : ACK byte if successful 15(hex) : NAK byte if unsuccessful
Description	This command draws an Empty/Wire-Frame polygon. Up to 7 vertices can be specified in any manner. Currently only a wire frame polygon is supported.	
Serial Example	Command Data: 67hex, 20hex, 05hex, 05hex, 20hex, 40hex, 28hex, 80hex, 10hex Draw a polygon.	

Draw Rectangle – **72hex**

Command	cmd, x1, y1, x2, y2, colour(msb:lsb)	
4DSL Cmd	Rectangle(x1, y1, x2, y2, colour)	
	cmd	72(hex) or r(ascii) : Command header byte
	x1	Top left horizontal start position of rectangle.
	y1	Top left vertical start position of rectangle.
	x2	Bottom right horizontal end position of rectangle.
	y2	Bottom right vertical end position of rectangle.
	colour	2 bytes define the rectangle colour.
Response	acknowledge	
	acknowledge	06(hex) : ACK byte if successful 15(hex) : NAK byte if unsuccessful
Description	This command will draw a coloured rectangle from point (x1, y1) to point (x2, y2) on the screen. If colour is chosen to be that of the background then the effect will be erasure. If Pen Size value was previously set to 0, the rectangle will be solid, otherwise it will be wire-frame if value was 1.	
Serial Example	Command Data: 72hex, 00hex, 00hex, 50hex, 50hex, 00hex, 1Fhex Draw a blue rectangle.	



Comandos de texto

Set Font – 46hex

Command	cmd, fontSet	
4DSL Cmd	Font(fontSet)	
	cmd	46(hex) or F(ascii) : Command header byte
	fontSet	Selects one of internal fonts. The supplied 3 fonts are: 00hex : 5x7 small size font set 01hex : 8x8 medium size font set 02hex : 8x12 large size font set These fonts can be altered and other fonts can be added.
Response	acknowledge	
	acknowledge	06(hex) : ACK byte if successful 15(hex) : NAK byte if unsuccessful
Description	This command selects one of the available internal fonts. Changes take place after the command is sent. Any character on the screen with the previous font set will remain as it was. NOTE: The GOLDELOX-SGX is shipped with three fonts displaying the characters 0x20 to 0x7f ' . i.e. Space to the character after the tilde. The use can alter the number of fonts, delete existing fonts, and, or, add extra fonts, up to the amount of available user flash (a very limited resource). A font does not need to start at 0x20, or end at 0x7f. It could, for example start at 0x30 ('0') and end at 0x39 ('9').	
Serial Examples	Command Data: 46hex, 00hex (Select small 5x7 font). Command Data: 46hex, 00hex (Select medium 8x8 font). Command Data: 46hex, 00hex (Select large 8x12 font).	



Draw Text Button – **62hex**

Command	cmd, state, x, y, buttonColour(msb:lsb), font, stringColour(msb:lsb), width, height, "string",	
4DSL Cmd	Button(state, x, y, buttonColour, font, stringColour, width, height, 'string')	
	cmd	62 (hex) or b (ascii) : Command header byte
	state	This byte specifies whether the displayed button is drawn UP (not pressed) or DOWN (pressed). 0 : Button Down (pressed) 1 : Button Up (not pressed)
	x	Top left horizontal start position of the button.
	y	Top left vertical start position of the button.
	buttonColour	2 bytes define the button colour.
	font	This byte specifies which internal font set to use for the string. The supplied fonts are: 0 : 5x7 internal font 1 : 8x8 internal font 2 : 8x12 internal font These fonts can be altered and other fonts can be added.
	stringColour	2 bytes define the string text colour.
	width	This byte defines the width or horizontal size (x magnification) of the character in the string. Effects the total width of the string and button.
	height	This byte defines the height or vertical size (y magnification) of the character in the string. Effects the total height of the string and button.
	"string"	String of ASCII characters displayed inside the button. Limit the string to a single line width.
	terminator	The string must be terminated with 00hex .
Response	acknowledge	
	acknowledge	06 (hex) : ACK byte if successful 15 (hex) : NAK byte if unsuccessful
Description	This command will place a Text button similar to the ones used in a PC Windows environment. The (x, y) refers to the top left corner of the button and the size of the button is automatically calculated and drawn on the screen with the string text relatively justified inside the button. The button can be displayed in an UP (button not pressed) or DOWN (button pressed) position by specifying the appropriate value in the 'state' byte. Separate button and text colours provide many variations in appearance and format.	



Draw ASCII Character (text format) – **54hex**

Command	cmd, char, column, row, charColour(msb:lsb)	
4DSL Cmd	AsciiChar(char, column, row, charColour)	
	cmd	54(hex) or T(ascii) : Command header byte
	char	Inbuilt standard ASCII character. range : 32dec – 127dec (20hex - 7Fhex).
	column	Horizontal position of the character (character units). range : 0 - 20 for 5x7 font. range : 0 - 15 for 8x8 and 8x12 fonts.
	row	Vertical position of the character (character units). range : 0 - 15 for 5x7 and 8x8 fonts. range : 0 - 9 for 8x12 font.
	charColour	2 bytes define the character colour.
Response	acknowledge	
	acknowledge	06(hex) : ACK byte if successful 15(hex) : NAK byte if unsuccessful
Description	This command will draw/display an ASCII character anywhere on the screen in character unit coordinates. The horizontal position of the character is specified by the column and The vertical position is specified by the row parameters.	
Serial Example	Command Data: 54hex, 41hex, 00hex, 00hex, FFhex, FFhex Draw/Display character 'A' (41hex) at column = 0, row = 0, colour = white (FFFFhex).	



Comandos de memory card (uSD)

Display Image-Icon from Memory Card - @49hex

Command	ext_cmd, cmd, x, y, width, height, colourMode, SectorAdd(hi:mid:lo)	
4DSL Cmd	UsdImage(x, y, width, height, colourMode, SectorAdd)	
	ext_cmd	40 (hex) or @ (ascii) : Extended Command header byte
	cmd	49 (hex) or I (ascii) : Command header byte
	x	Image horizontal start position (top left corner).
	y	Image vertical start position (top left corner).
	width	Horizontal size of the image.
	height	Vertical size of the image.
	colourMode	10 (hex) : 65K colour mode, 16bits/2bytes per pixel .
	SectorAdd	3 bytes (big endian) sector address of a previously stored Image-Icon that is about to be displayed.
Response	acknowledge	
	acknowledge	06 (hex) : ACK byte if successful 15 (hex) : NAK byte if unsuccessful
Description	<p>This command displays a bitmap image or an icon on the screen that has been previously stored at a particular sector address in the memory card. The screen position of the image to be displayed is specified by (x, y) and the size of the image by width and height parameters.</p> <p>The colourMode byte parameter can only be set to 10hex, i.e. the previously stored image can only be 16 bit colour format (2 bytes per pixel).</p> <p>Notes:</p> <ul style="list-style-type: none"> • The “Screen Copy-Save to Memory Card” command always stores that part of the screen as a 16 bit image, i.e. 2 bytes per pixel. • Do not store an image/icon in 8 bit colour format, this will result in a corrupted image. • The images or icons when stored into the memory card must be sector boundary aligned, i.e. the object start location must be at the start of a sector boundary. 	
Serial Example	<p>Command Data: 40hex, 49hex, 00hex, 00hex, 80hex, A0hex, 10hex, 3Bhex, 16hex, 04hex</p> <p>Display the image at (0, 0) from sector address 3B1604hex.</p>	



Display Video-Animation Clip from Memory Card - @56hex

Command	ext_cmd, cmd, x,y,width, height, colourMode, delay, frames(msb:lsb),	
4DSL Cmd	Video(x, y, width, height, colourMode, delay, frames, SectorAdd)	
	ext_cmd	40(hex) or @(ascii) : Extended Command header byte
	cmd	56(hex) or V(ascii) : Command header byte
	x	Video horizontal start position (top left corner).
	y	Video vertical start position (top left corner).
	width	Horizontal size of the video-animation.
	height	Vertical size of the video-animation.
	colourMode	10(hex) : 65K colour mode, 16bits/2bytes per pixel .
	delay	1 byte inter-frame delay in milliseconds.
	frames	2 bytes (big endian) total frame count in the video-animation
	SectorAdd	3 bytes (big endian) sector address of a previously stored video-animation clip that is about to be displayed.
Response	acknowledge	
	acknowledge	06(hex) : ACK byte if successful 15(hex) : NAK byte if unsuccessful
Description	<p>This command plays a video or an animation clip on the screen that has been previously stored at a particular sector address in the memory card. The screen position of the clip to be played is specified by (x, y) and the size of the clip by width and height parameters.</p> <p>The colourMode byte parameter can only be set to 10hex, i.e. the previously stored video/animation can only be 16 bit colour format (2 bytes per pixel).</p> <p>Notes:</p> <ul style="list-style-type: none"> Do not store a video/animation in 8 bit colour format, this will result in a corrupted result. 	
Serial Example	Command Data: 40hex, 56hex, 00hex, 00hex, 80hex, A0hex, 10hex, 01hex, 64hex, 00hex, 00hex, 04hex	

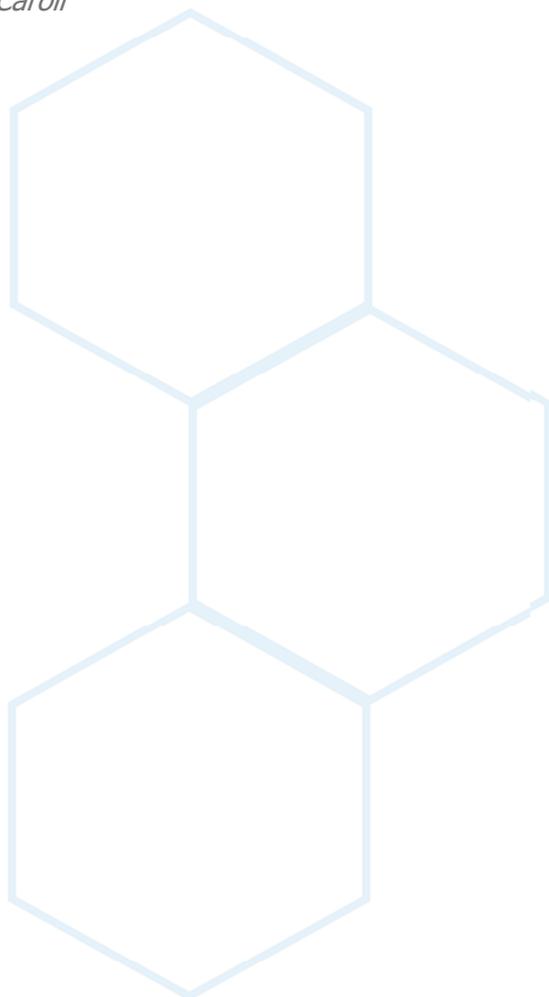


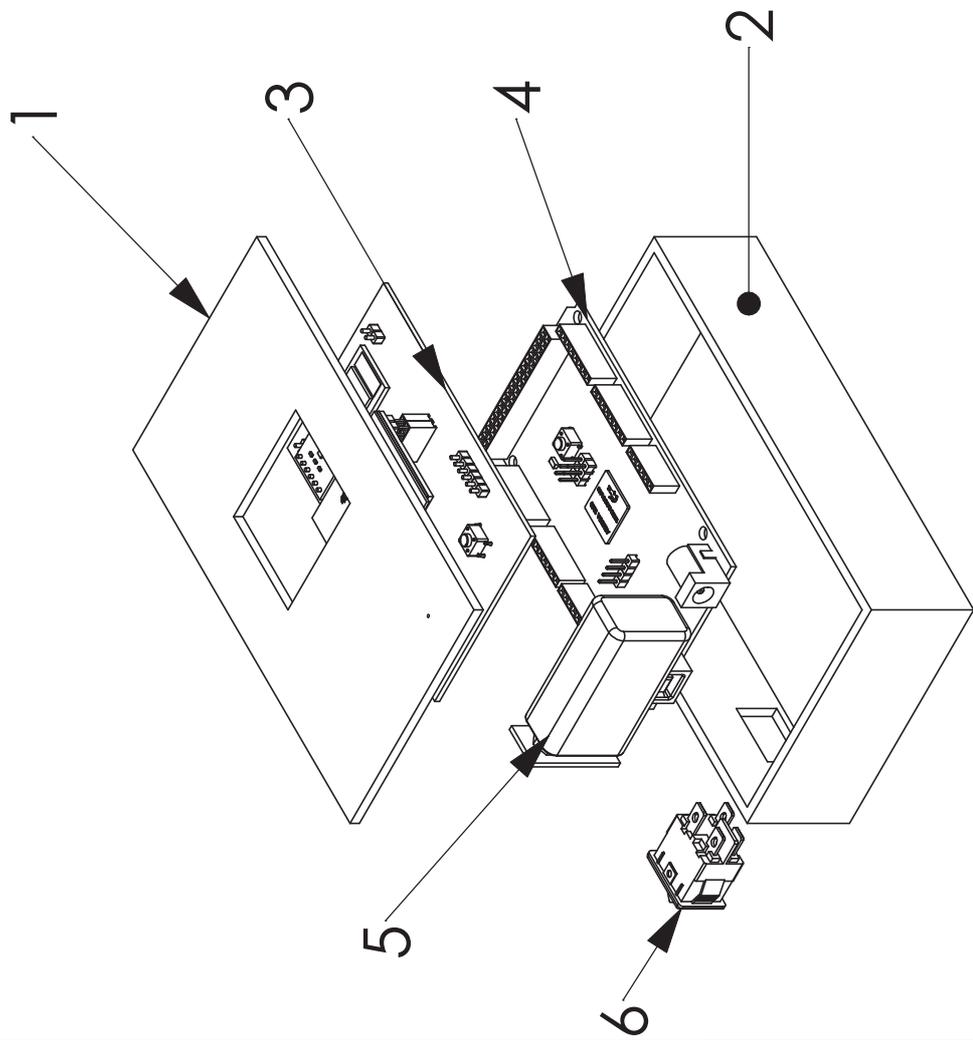
PLANOS

Apéndice E

"Agotemos lo posible, intentemos lo imposible"

-Lewis Carroll



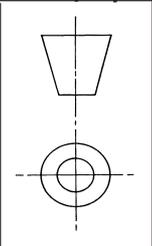


6	SWITCH	*****	1
5	BATERIA	9 V	1
4	ARDUINO	MEGA 2560	1
3	ENSAMBLE PLACA	STI-EPS-700-C	1
2	CAJA	*****	1
1	TAPA SUPERIOR	STI-PTS-504-C	1
Nº	PIEZA	REFERENCIA	CANT

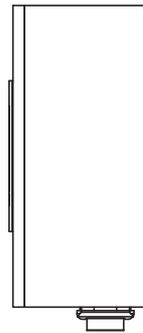
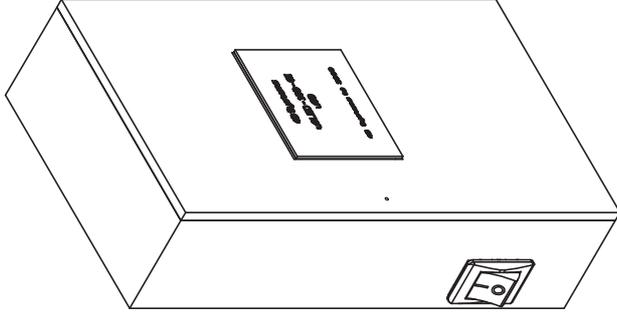
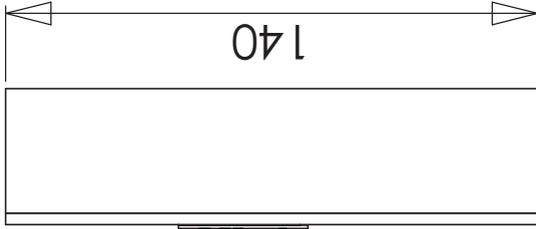
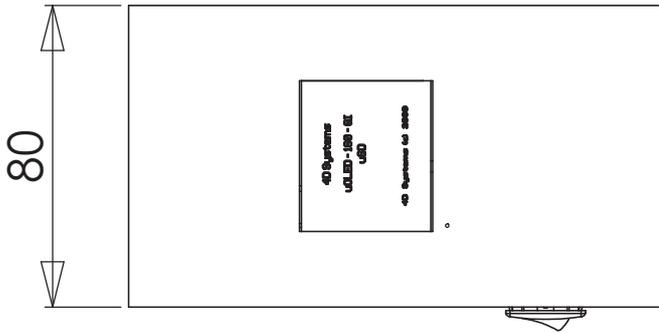


Centro de Diseño Mecánico e Innovación
Tecnológica

Título:
Ensamble
General

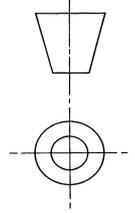


Rev: A	Escala: 1:2
Fecha: 30/07/2012	Acot: [mm]
Dibujó: J.C.R.	Cantidad: 1
Revisó: P.M.D	MATERIAL:
Aprobó: S.C.C.	Multiples
Dib. No. STI-EGS-500-C	



Centro de Diseño Mecánico e Innovación Tecnológica

Título: Sistema de Tablero de Instrumentos



Rev: A

Fecha: 30/07/2012

Escala: 1:2

Acot: [mm]

Dibujó: J.C.R.

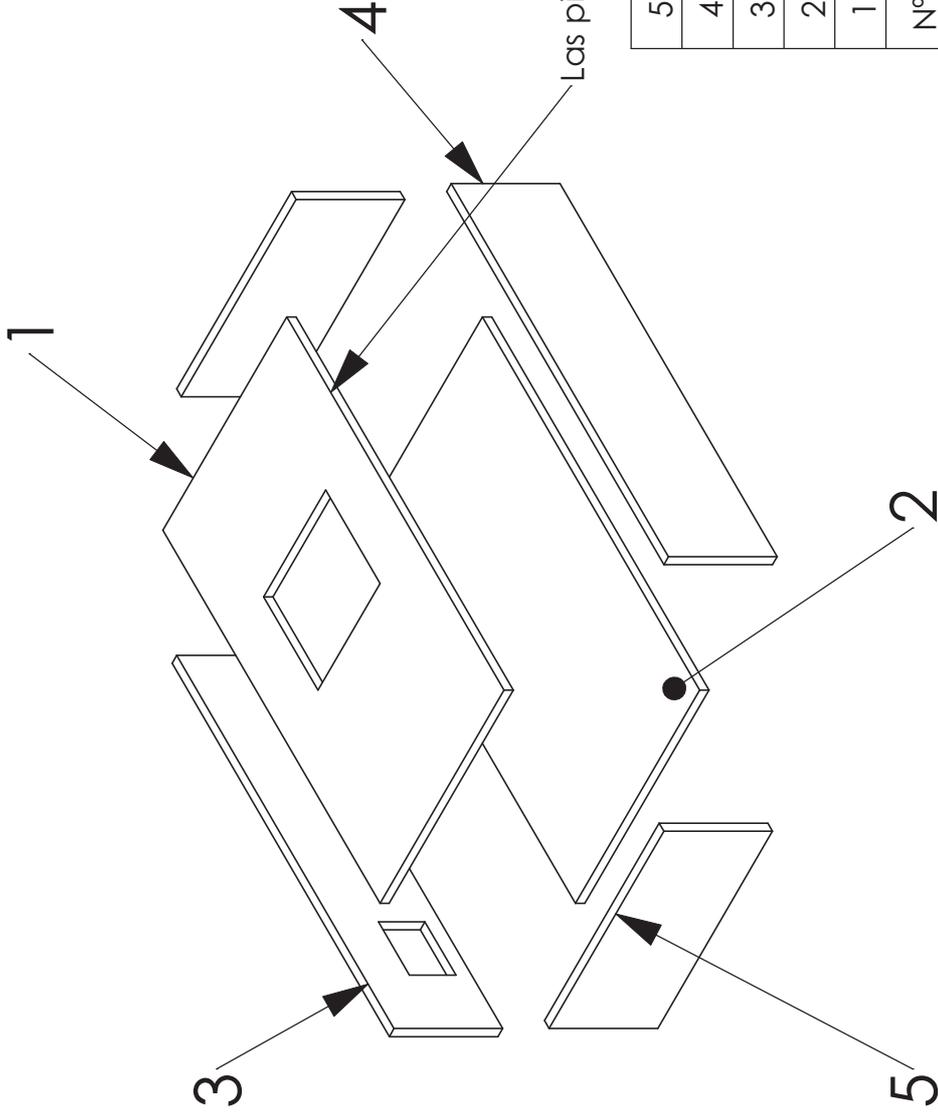
Revisó: P.M.D

Cantidad: 1

MATERIAL: Múltiples

Aprobó: S.C.C.

Dib. No. STI-STI-501-C



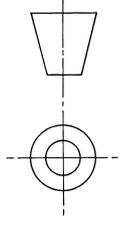
Las piezas 1,2,3,4 y 5 llevan una union adhesiva simple a tope

5	COSTADO C	STI-PCC-508-C	2
4	COSTADO B	STI-PCB-507-C	1
3	COSTADO A	STI-PCA-506-C	1
2	TAPA INFERIOR	STI-PTI-505-C	1
1	TAPA SUPERIOR	STI-PTS-504-C	1
Nº	PIEZA	REFERENCIA	CANT



Centro de Diseño Mecánico e Innovación
Tecnológica

Título:
Ensamble
General Caja



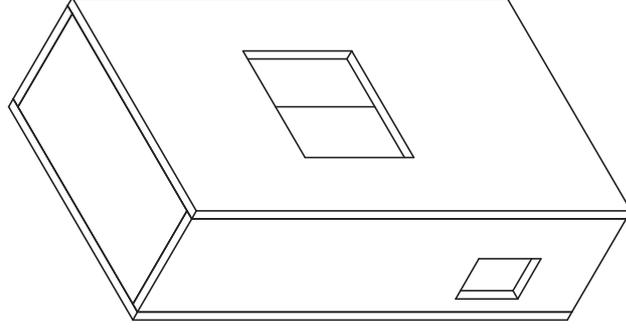
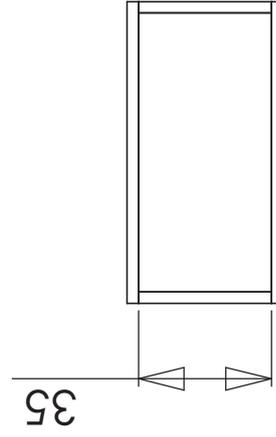
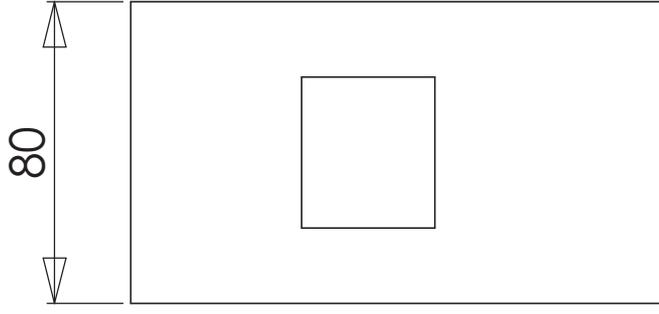
Rev: A Escala: 1:2

Fecha: 30/07/2012 Acot: [mm]

Dibujó: J.C.R. Cantidad: 1

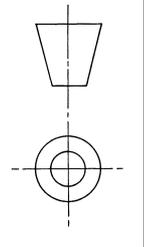
Revisó: P.M.D MATERIAL:
Aprobó: S.C.C. Acrílico

Dib. No. STI-EGC-502-C

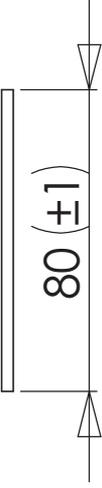
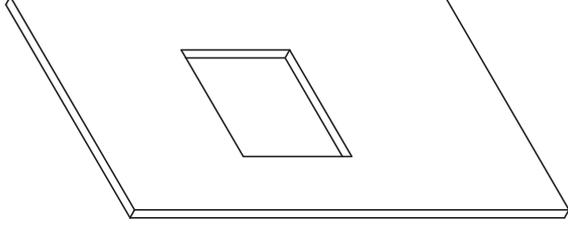
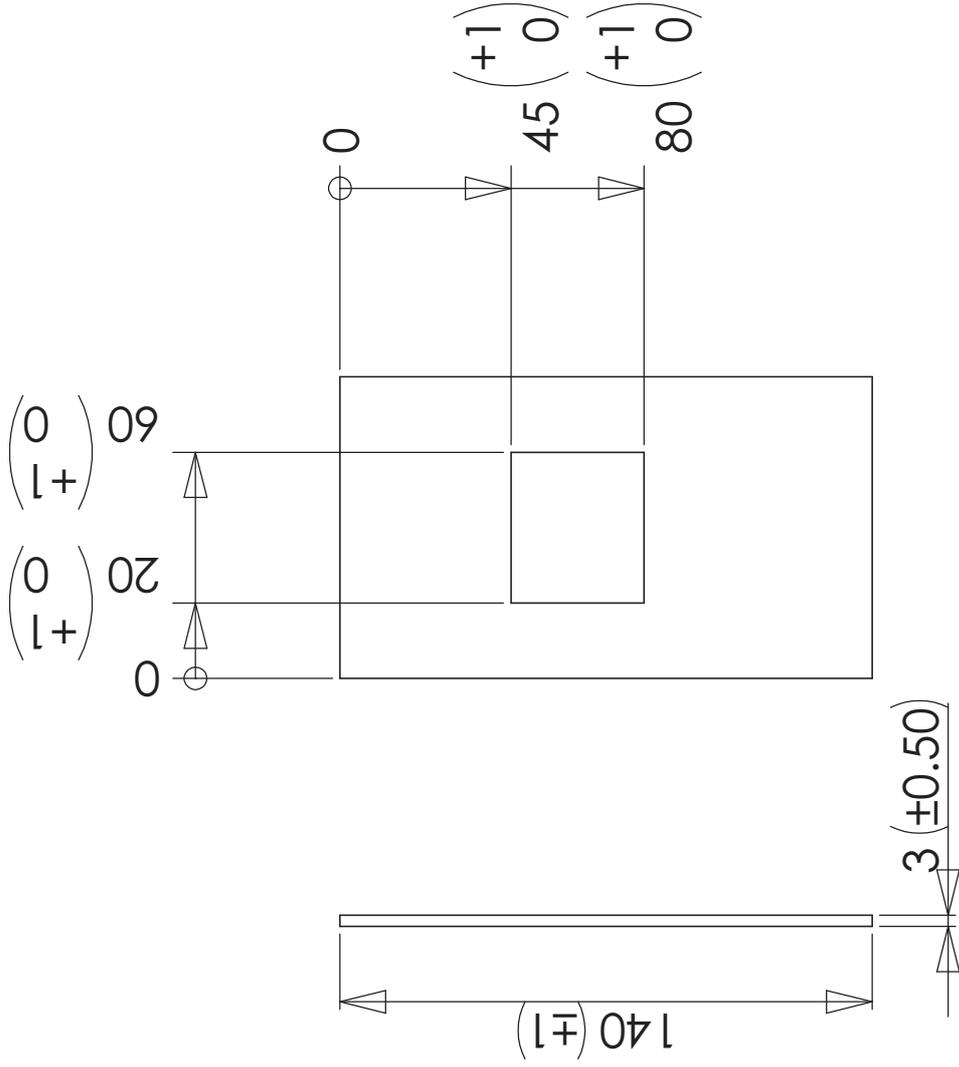


Centro de Diseño Mecánico e Innovación Tecnológica

Título: Caja del Sistema de Tablero

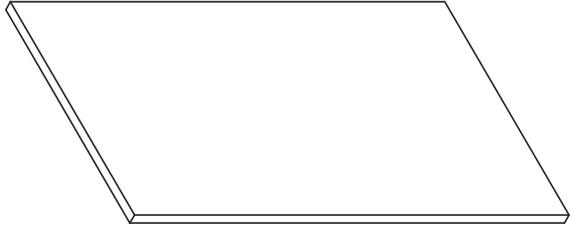
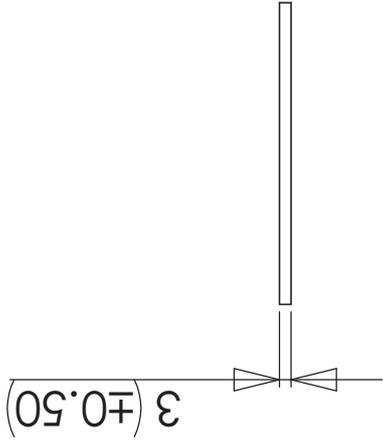
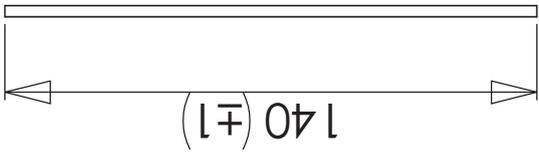
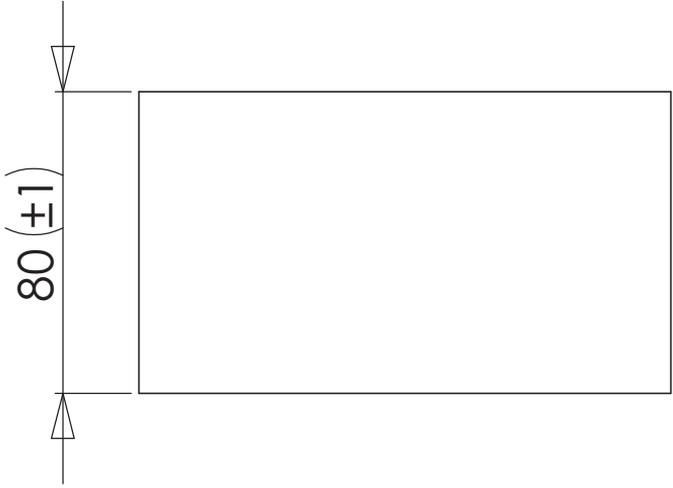


Rev: A	Escala: 1:2
Fecha: 30/07/2012	Acot: [mm]
Dibujó: J.C.R.	Cantidad: 1
Revisó: P.M.D	MATERIAL: Acrílico
Aprobó: S.C.C.	
Dib. No. STI-CST-503-C	



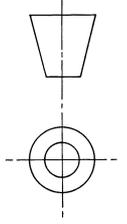
Centro de Diseño Mecánico e Innovación Tecnológica			
Título: Tapa Superior			

Rev: A	Escala: 1:2
Fecha: 30/07/2012	Acot: [mm]
Dibujó: J.C.R.	Cantidad: 1
Revisó: P.M.D	MATERIAL: Acrílico
Aprobó: S.C.C.	
Dib. No. STI-PTS-504-C	



Centro de Diseño Mecánico e Innovación
Tecnológica

Título: Tapa
Inferior



Rev: A

Fecha: 30/07/2012

Escala: 1:2

Acot: [mm]

Dibujó: J.C.R.

Revisó: P.M.D

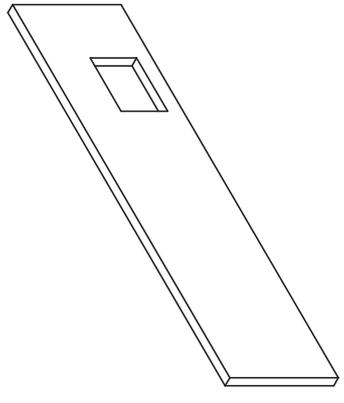
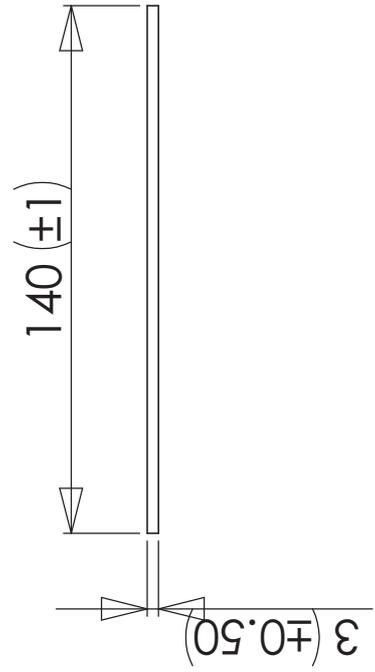
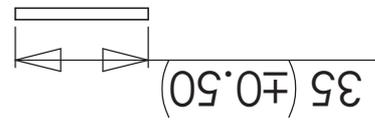
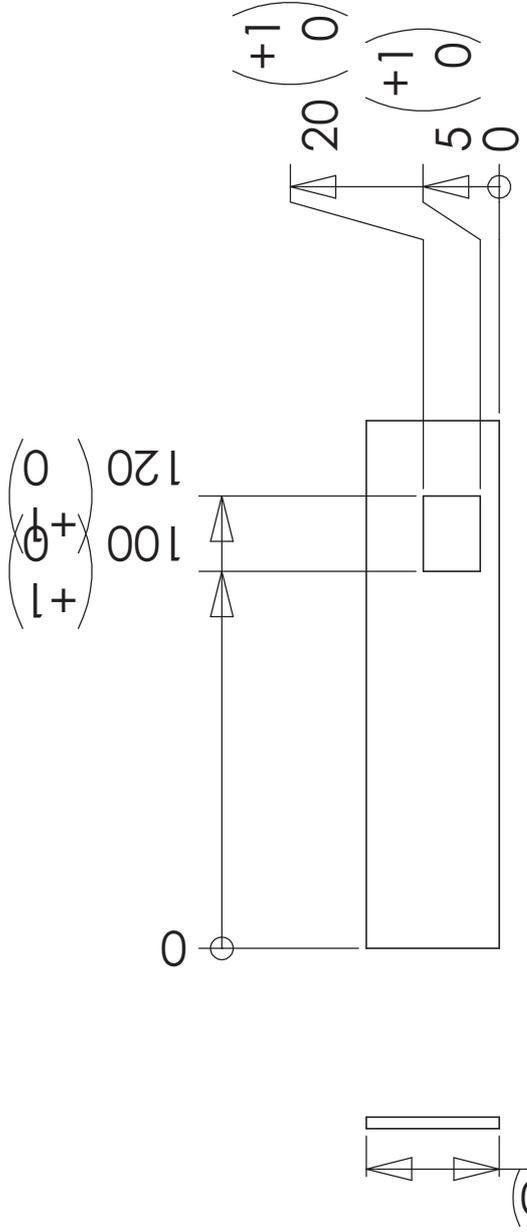
Aprobó: S.C.C.

Cantidad: 1

MATERIAL:

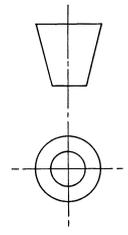
Acrílico

Dib. No. STI-PTI-505-C

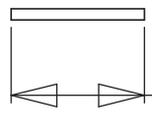
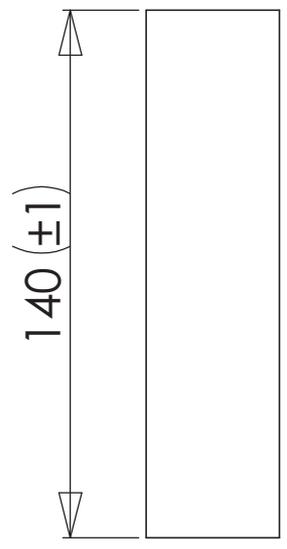
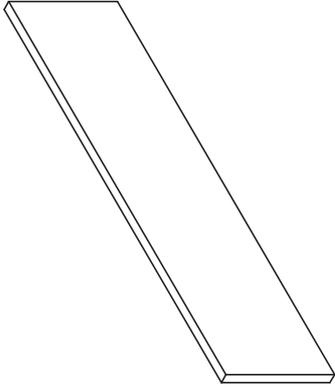


Centro de Diseño Mecánico e Innovación Tecnológica

Título:
Costado A

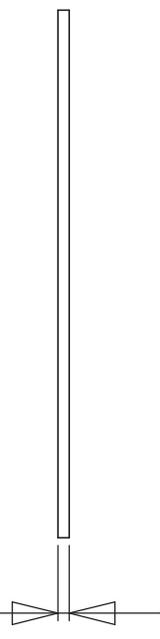


Rev: A	Escala: 1:2
Fecha: 30/07/2012	Acot: [mm]
Dibujó: J.C.R.	Cantidad: 1
Revisó: P.M.D	MATERIAL: Acrylic
Aprobó: S.C.C.	
Dib. No. STI-PCA-506-C	

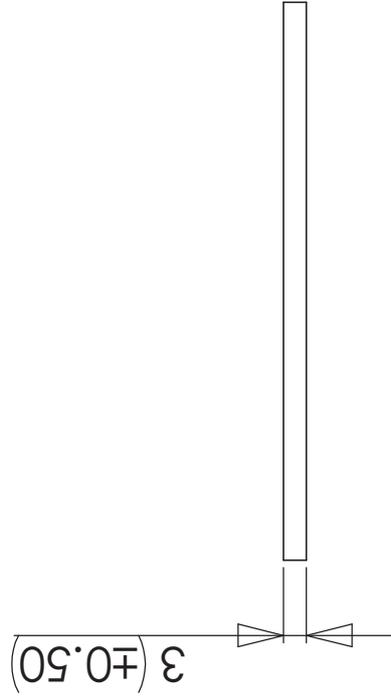
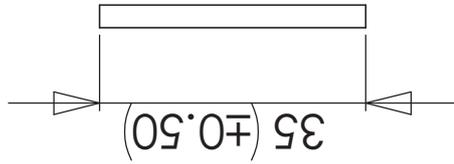
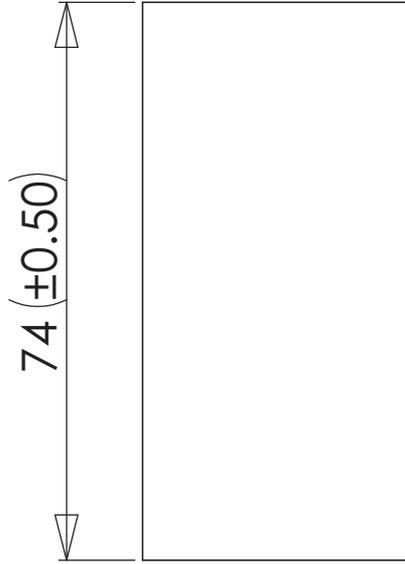
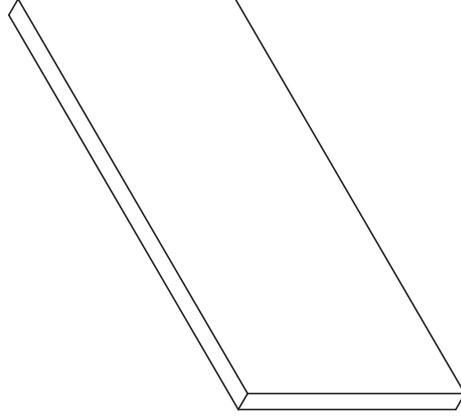


35 (±0.50)

3 (±0.50)

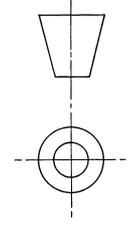


Centro de Diseño Mecánico e Innovación Tecnológica	Rev: A	Escala: 1:2
		Fecha: 30/07/2012
Título: Costado B	Dibujó: J.C.R.	Cantidad: 1
	Revisó: P.M.D	MATERIAL: Acrylic
	Aprobó: S.C.C.	
	Dib. No. STI-PCB-507-C	



Centro de Diseño Mecánico e Innovación
Tecnológica

Título:
Costado C



Rev: A

Fecha: 30/07/2012

Escala: 1:1

Acot: [mm]

Dibujó: J.C.R.

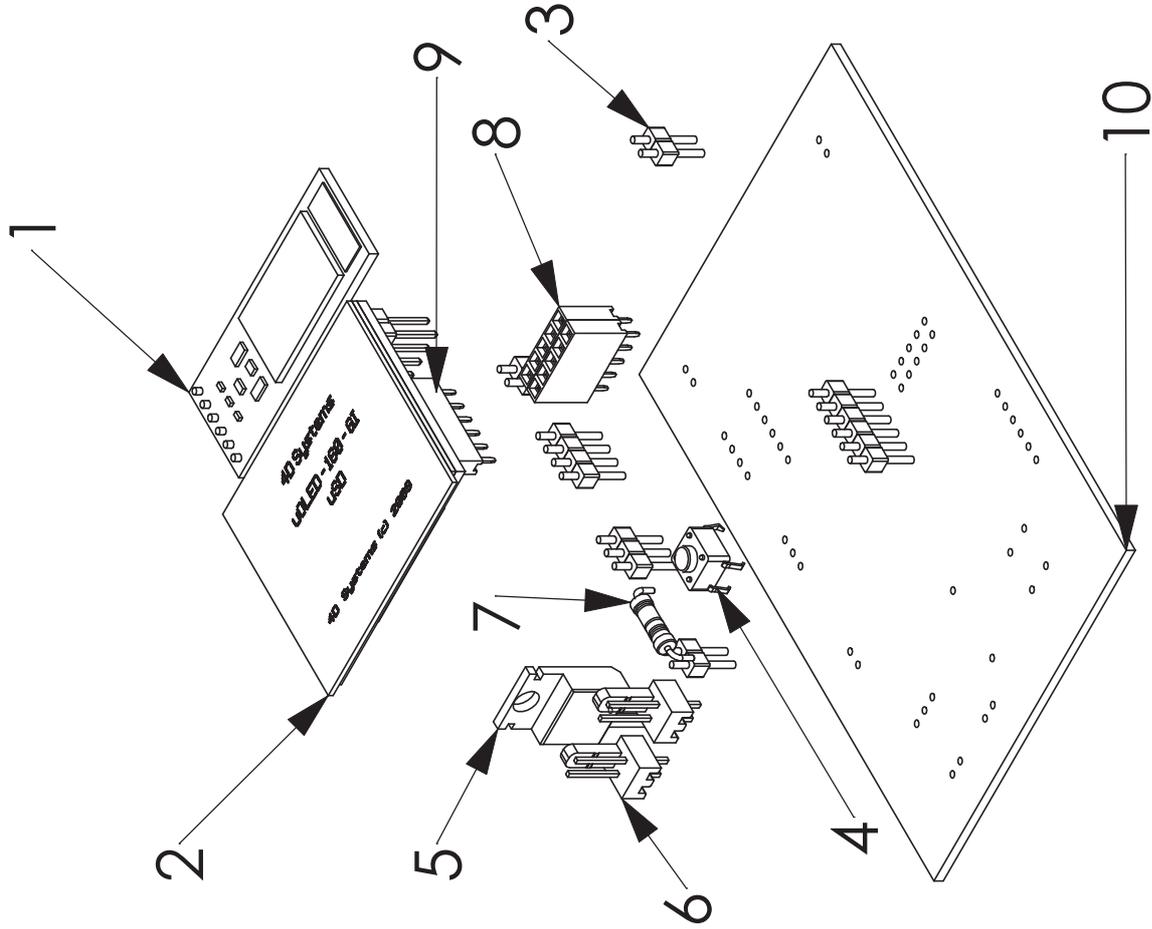
Cantidad: 2

Revisó: P.M.D

MATERIAL:
Acrylic

Aprobó: S.C.C.

Dib. No. STI-PCC-508-C

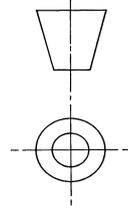


10	PCB	STI-PCB-702-C	1
9	HEADER	HEMBRA 6 PINES	1
8	HEADER	HEMBRA 5 PINES	2
7	RESISTOR	100 OHMS	1
6	MOLEX	MACHO 2 PINES	2
5	REGULADOR	7805	1
4	BUTTON	*****	1
3	HEADER	MACHO	19
2	PANTALLA	UOLED-160	1
1	BLUETOOTH	BLUESMIRF-GOLD	1
Nº	PIEZA	REFERENCIA	CANT

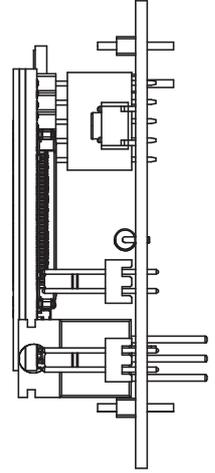
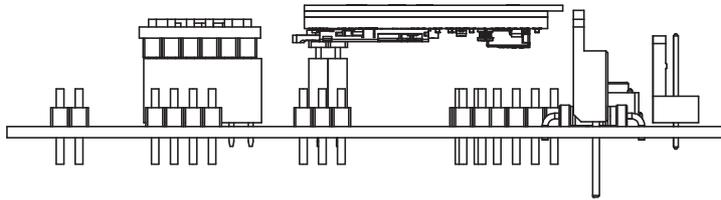
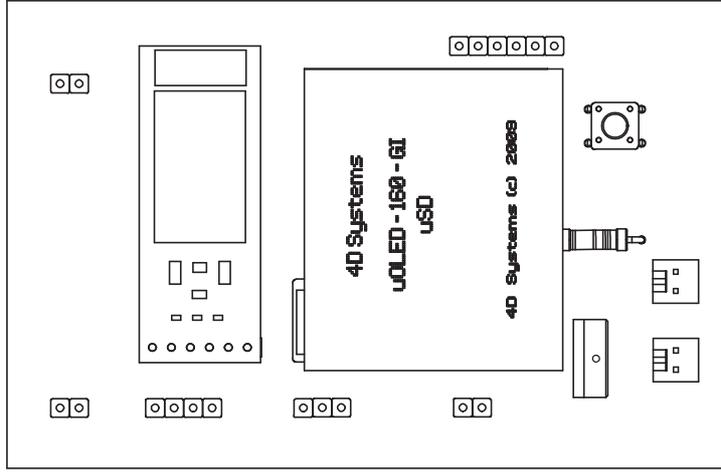
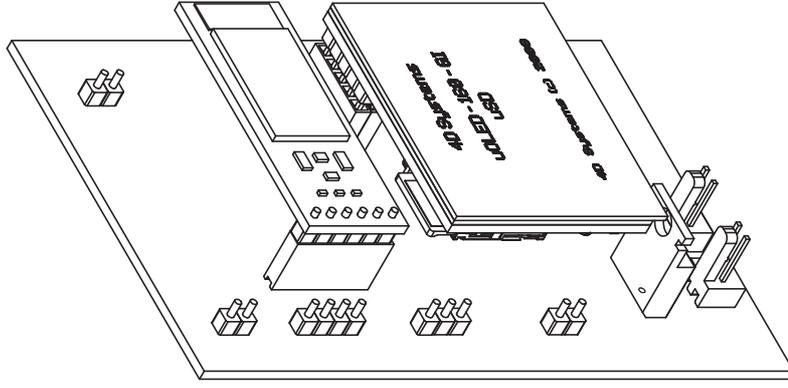


Centro de Diseño Mecánico e Innovación
Tecnológica

Título:
Ensamble Placa
del Sistema

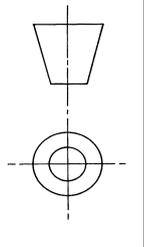


Rev: A	Escala: 1:1
Fecha: 30/07/2012	Acot: [mm]
Dibujó: J.C.R.	Cantidad: 1
Revisó: P.M.D	MATERIAL:
Aprobó: S.C.C.	Multiple
Dib. No. STI-EPS-700-C	

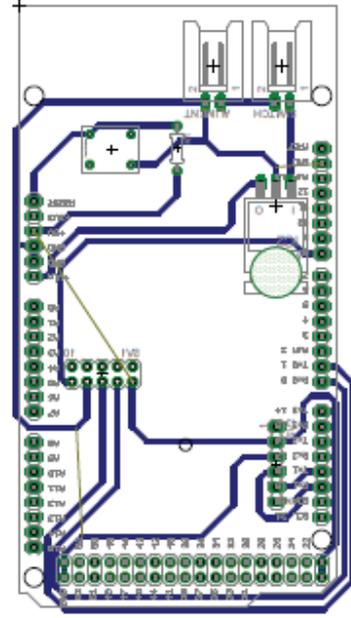


Centro de Diseño Mecánico e Innovación Tecnológica

Título: Placa Sistema tablero de Instrumentos



Rev: A	Escala: 1:1
Fecha: 30/07/2012	Acot: [mm]
Dibujó: J.C.R.	Cantidad: 1
Revisó: P.M.D	MATERIAL: Múltiples
Aprobó: S.C.C.	
Dib. No. STI-PST-701-C	



	Centro de Diseño Mecánico e Innovación Tecnológica		Rev. A Fecha: 30/07/2013	Escala: 1:1 Acot: [mm]
	Título: PCB		Dibujo: J.C.R. Revisó: P.M.D Aprobó: S.C.C.	Cantidad: 1 MATERIAL: Cobre
			Dib. No. STI-PCB-702-C	

U1 ATMEGA2560

