





Instituto Tecnológico y de Estudios Superiores de Monterrey  
Campus Ciudad de México

Escuela de Diseño; Ingeniería y Arquitectura  
Departamento de Ingeniería Mecatrónica  
Ingeniería en Telecomunicaciones y Sistemas Electrónicos

# **Sistema de Adquisición de Datos y Telemetría para un *Kart***

Autores:

Mónica Mimendi Guerrero A01124748

Andrés Muñoz Reskala A01124019

Marcelo Leon Ayala A01124746

Asesor:

Dr. Hugo Ramón Elizalde Siller

Revisado por:

Dr. Martín Rogelio Bustamante



## Resumen

Este documento sirve de guía para la mejor comprensión del proyecto. Se detallan los pasos que se siguieron para todo el desarrollo desde el proceso de recopilación de información y toma de decisiones hasta la implementación y análisis de resultados.

El documento consta de varios capítulos, en el capítulo uno es la introducción al proyecto, en él se mencionan los antecedentes, la problemática a la que nos enfrentamos y los objetivos que nos planteamos; en el capítulo dos se incluye el estado del arte y un marco teórico que explica el funcionamiento de las tecnologías y dispositivos usados; el capítulo tres es la descripción de cómo se desarrolló el proyecto, incluye los diseños de *hardware* y *software*, con diagramas a bloques y diagramas de flujo; en el capítulo cuatro se incluyen todas las pruebas y resultados obtenidos para finalmente aportar nuestras propias conclusiones y posibles mejoras al sistema en el capítulo cinco del documento.

El tiempo que duró el desarrollo de este proyecto de ingeniería fue de un año y estuvo dividido en dos partes, en la primera parte nos enfocamos a realizar la investigación pertinente, definir objetivos, establecer el material necesario y desarrollar la adquisición de datos, se realizaron unas pruebas de envío de datos bajo un protocolo de comunicación, sin embargo posteriormente se optó por otro protocolo. En la segunda parte nos dedicamos a la implementación del sistema de telemetría y del nuevo protocolo para la comunicación inalámbrica dentro del vehículo y se desarrollaron diferentes aplicaciones. Finalmente se unen las dos partes para obtener un resultado final.

Este documento también cuenta con una sección dedicada a los anexos, en estos se ubican los códigos realizados en diversos programas y lenguajes, son los que hacen funcionar al proyecto.



## Índice

Acrónimos .....	11
.....	15
1.1 Introducción .....	15
1.2 Problemática .....	16
1.3 Justificación .....	17
1.4 Objetivos .....	17
1.4.1 Objetivo general .....	17
1.4.2 Objetivos Específicos .....	18
1.5 Alcances y limitaciones .....	20
<b>Capítulo 2. Marco teórico</b> .....	<b>21</b>
2.1 Antecedentes .....	21
2.2 Marco teórico.....	23
2.2.1 Comunicación <i>Wi-Fi</i> .....	23
2.2.2 Comunicación Serial .....	26
2.2.3 EVK1100 .....	28
2.2.4 Tarjeta SD .....	31
2.2.5 RCM5400W .....	33
2.2.6 iPad.....	34
<b>Capítulo 3. Desarrollo del Sistema</b> .....	<b>35</b>
3.1 Descripción general del sistema .....	35
3.2 Diseño del prototipo .....	37
3.2.1 Sistema de adquisición de datos .....	37
3.2.2 Sistema de alarmas .....	43
3.2.3 Telemetría .....	47
3.3 Diseño del programa ( <i>software</i> ) .....	50
3.3.1 Programación del microcontrolador UC3A0512 para la adquisición de datos y el almacenamiento de información en la tarjeta SD .....	50

3.3.2 Programación del módulo RS232/Wi-Fi.....	54
3.3.3 Programación del sistema de archivos para lectura de tarjeta SD .....	57
3.3.4 Programación de la interfaz gráfica para los dispositivos móviles .....	58
3.4 Alternativas de Solución.....	60
<b>Capítulo 4. Pruebas y resultados.....</b>	<b>61</b>
4.1 Recepción de datos por serial .....	61
4.2 Pruebas del muestreador.....	63
4.3 Escritura de datos en la tarjeta SD.....	65
4.5 Prueba de la lectura de la SD .....	68
4.6 Pruebas de despliegue de gráficas en pits .....	70
4.7 Resumen de las pruebas realizadas .....	74
<b>Capítulo 5. Conclusiones .....</b>	<b>77</b>
5.1 Conclusiones .....	77
5.2 Trabajo a futuro .....	78
<b>Bibliografía y referencias.....</b>	<b>79</b>
Bibliografía .....	79
Referencias electrónicas .....	79
<b>Anexos .....</b>	<b>81</b>
Simulación del funcionamiento de alarmas en Matlab .....	82
Código del microcontrolador externo.....	90

## Índice de tablas

Tabla 1. Descripción de pines del serial .....	27
Tabla 2. Características de los tipos de tarjetas SD [16] .....	32
Tabla 3. Medidas de la llanta .....	42
Tabla 4 Valores máximos de parámetros.....	43
Tabla 5. Tablas de reglas del sistema difuso .....	45
Tabla 6. Rangos de operación para rpm y potencia.....	46
Tabla 7. Valores para la defusificación.....	46
Tabla 8. Descripción de las funciones del sistema difuso .....	47
Tabla 9. Configuraciones de los módulos del UC3A0512.....	53
Tabla 10. Descripción de parámetros de configuración para <i>Wi-Fi</i> .....	54



## Índice de imágenes

Imagen 1. Diagrama de una red por Wi-Fi [9].....	25
Imagen 2. Punto de Acceso.....	25
Imagen 3. Red inalámbrica Wi-Fi [9].....	26
Imagen 4. Pines del puerto serial [13].....	27
Imagen 5. EVK1100.....	28
Imagen 6. Diagrama a Bloques de USART [14].....	30
Imagen 7. Diagrama a Bloques de la tarjeta RCM5400W [17].....	33
Imagen 8. Diagrama a Bloques del Sistema.....	35
Imagen 9. Diagrama a Bloques del Sistema de Adquisición de Datos.....	38
Imagen 10 Trama Sincronía.....	39
Imagen 11. Diagrama a Bloques de la medición de los rpm. <i>Rueda</i> [19], <i>sensor</i> [20], <i>Schmitt Trigger</i> [21].....	40
Imagen 12. Circuito digitalizador de señal.....	41
Imagen 13. Respuesta del Schmitt Trigger [21].....	41
Imagen 14. Interfaz gráfica del iPad (tablero).....	43
Imagen 15. Espacio difuso para los rpm del motor.....	44
Imagen 16. Espacio difuso para la potencia entregada por el motor.....	44
Imagen 17. Gráfica de defusificación.....	45
Imagen 18. Diagrama a bloques del sistema de telemetría.....	49
Imagen 19. Interfaz gráfica de los dispositivos móviles de pits.....	50
Imagen 20. Diagrama de flujo.....	51
Imagen 21. Diagrama de flujo de la recepción de datos de la etapa de control.....	52
Imagen 22. Envío de datos al módulo RS232/Wi-Fi.....	53
Imagen 23. Diagrama de flujo del programa principal del rabbit 5000.....	56
Imagen 24. Interfaz gráfica de la aplicación para lectura de tarjeta SD.....	57
Imagen 25. Diagrama de flujo de la aplicación web para pits.....	59
Imagen 26. Diagrama del envío de trama entre etapa de control y tarjeta de adquisición de datos.....	61
Imagen 27. Prueba de la recepción por serial en la EVK1100.....	62

Imagen 28 Código de configuración del puerto serial .....	63
Imagen 29. Diagrama a bloques de la conexión para la prueba del muestreador .....	63
Imagen 30. a) Respuesta del digitalizador .....	64
Imagen 31. b) Respuesta del digitalizador .....	64
Imagen 32. Recepción por serial de la velocidad de la llanta .....	64
Imagen 33. Lectura de un sector de la tarjeta SD desde el programa HxD .....	66
Imagen 34. Identificación de dispositivo bluetooth.....	64
Imagen 35. Identificación de dispositivo bluetooth.....	67
Imagen 36. Inserción de la tarjeta SD en la laptop .....	68
Imagen 37. Abrir aplicación para leer la tarjeta SD.....	68
Imagen 38. Interfaz gráfica de la aplicación para leer tarjeta SD .....	69
Imagen 39. Búsqueda y selección de carpeta destino .....	69
Imagen 40. Seleccionar unidad en disco para leer.....	69
Imagen 41. Pantalla principal y registro del usuario.....	70
Imagen 42. Despliegue del menú.....	70
Imagen 43. Selección de parámetros a graficar .....	71
Imagen 44. Despliegue de gráficas.....	71



## Sistema de Adquisición de Datos y Telemetría para un *Kart*

## Acrónimos

En esta sección se define la nomenclatura o palabras clave que es importante conocer para comprender mejor el contenido del documento.

### A

ADC *Analog To Digital Converter*  
AP *Access Point*

### B

BSS *Base Station Subsystem*

### C

cm *Centímetros*  
CTS *Clear To Send*

### D

DCE *Data Communication Equipment*  
DSP *Digital Signal Processing*  
DTE *Data Terminal Equipment*  
DTR *Data Terminal Ready*  
DS *Distribution System*  
DSR *Data Set Ready*

### E

EVK *Evaluation Kit*

### G

GHz *Giga Hertz*  
GPIO *General-Purpose Input / Output*  
GPRS *General Packet Radio Service*  
GPS *Global Positioning System*

### H

HTML5 *HyperText Markup Language version 5*  
Hz *Hertz*

### I

INTC *Interrupt Controller*

### J

JTAG *Joint Test Action Group*

### K

KB *Kilo Bytes*  
Kbps *Kilo bits per second*

**KHz**     *KiloHertz*

**L**

**LAN**     *Local Area Network*

**LCD**     *Liquid Crystal Display*

**LED**     *Light-Emitting Diode*

**M**

**MB**     *Mega Bytes*

**Mbps**    *Mega bits per second*

**MHz**     *Mega Hz*

**MMC**    *MultiMedia Card*

**MPEG**   *Moving Picture Experts Group*

**ms**     *Miliseconds*

**P**

**PCI**     *Peripheral Component Interconnect*

**PDA**     *Personal Digital Assistant*

**PHP**     *Hypertext Pre-Processor*

**PLL**     *Phase-Locked-Loop*

**PM**     *Power Manager*

**R**

**RGB**     *Red, Green, Blue*

**RISC**    *Reduced Instruction Set Computer*

**RTS**     *Request To Send*

**RxD**     *Received Data*

**S**

**SPI**     *Serial Peripheral Interface*

**STA**     *Stations*

**T**

**TC**     *Timer Counter*

**TxD**     *Transmit Data*

**U**

**UART**   *Universal Asynchronous Receiver-Transmitter*

**USART**   *Universal Synchronous/Asynchronous Receiver/Transmitter*

**USB**     *Universal Serial Bus*

**W**

**Wi-Fi**   *Wireless Fidelity*

**WLAN**   *Wireless Local Area Network*

**X**

**XML**     *Extensible Markup Language*





Sistema de Adquisición de Datos y Telemetría para un *Kart*

# Capítulo 1. Introducción al proyecto

---

## 1.1 Introducción

La idea del proyecto surge del deseo de cuatro alumnos de Ingeniería Mecánica del Tecnológico de Monterrey Campus Ciudad de México en participar en la competencia *Shell Eco-Marathon*, la cual se realiza en Estados Unidos de América.

La *Shell Eco-marathon* es una carrera de autos basados en fuentes alternas de energía. Formar parte en este evento reta a alumnos de secundaria y los equipos de estudiantes universitarios de todo el mundo para diseñar, construir y probar los vehículos energéticamente eficientes. Con eventos anuales en las Américas, Europa y Asia, los ganadores son los equipos que van la mayor distancia usando la menor cantidad de energía.[1] En este año (2011) la competencia será efectuada en la ciudad de Houston, Texas.

Para poder crear un prototipo funcional fue necesario integrar varios equipos de estudiantes de distintas carreras: Ingeniería Mecánica Eléctrica, Ingeniería Mecatrónica e Ingeniería en Telecomunicaciones y Sistemas Electrónicos. El equipo completo que representará a la Institución recibió de nombre de *Clean Mobilities*.

Por lo tanto nuestra labor dentro del equipo *Clean Mobilities* y como estudiantes de la carrera de Ingeniería en Telecomunicaciones y Sistemas Electrónicos fue desarrollar un sistema que permitiera monitorear y adquirir los datos relevantes del vehículo para, de alguna manera, desplegarlos en el tablero del carro y posteriormente leerlos en *pits*, esto con la finalidad de tener mejor desempeño durante la carrera.

La competencia se subdivide en varias categorías, una de ellas de nuestro interés: Innovación Tecnológica, la cual se basa en el uso de dispositivos y tecnologías novedosas para la resolución de un problema.



## 1.2 Problemática

Como se explicó en el punto 1.1 de este capítulo, como parte de la división de electrónica en el equipo *Clean Mobilities*, debemos realizar un sistema de adquisición de datos y de telemetría que ayude al equipo a tener un mejor desempeño en la carrera, además que pueda ser merecedor de un premio en la categoría de Innovación Tecnológica.

Por lo tanto el problema se reduce a diseñar un sistema que permita conocer el comportamiento del vehículo basándose en parámetros adquiridos del motor y de la celda de hidrógeno (que es la que se utiliza con el fin de sustituir al combustible). El conductor del vehículo debe poder visualizar gráficamente los parámetros adquiridos para saber el estado del carro, por cuestiones de seguridad y para la toma de decisiones adecuadas durante la carrera.

Por otro lado es necesario resolver el problema de la telemetría, es decir, cómo el equipo de personas ubicado en *pits* va a tener conocimiento del estado del vehículo. Se debe pensar en medios de comunicación entre el vehículo y *pits*. La manera de despliegue de la información en *pits* debe ser amigable con el usuario y fácil de manipular para poder fomentar un análisis rápido de la información y la toma de decisiones adecuadas, las cuales permitan al vehículo ganar la competencia o por lo menos quedar en un buen lugar.

Es importante investigar y seleccionar las tecnologías e instrumentación a emplear para facilitar la adquisición de datos y comunicación dentro del vehículo y hacia *pits*, así como el acceso fácil a la información del vehículo para poder conocer su eficiencia.

También es indispensable señalar que no sólo debemos pensar en la manera de dar resolución a este problema, sino que tenemos que cumplir con ciertas especificaciones y reglas que *Shell* (la empresa organizadora de la carrera) establece.

Algunas de las reglas que nos son de interés son las siguientes:

De acuerdo al artículo 71 del reglamento de la competencia solo se permite usar una batería por vehículo para la electrónica del carro, no se permiten otras fuentes de energía adicionales; los dispositivos que pueden usar baterías adicionales son los sistemas de radiocomunicación, sistemas GPS (*Global Positioning System*) y ventiladores. Finalmente en caso de usar pilas de Litio se debe instalar un Sistema de monitoreo de batería para controlar y protegerla en caso de riesgos o incendios. Por otro lado, en el artículo 17 del mismo reglamento se especifica que queda prohibida la comunicación que implique que el conductor deba sostener un dispositivo, el conductor debe permanecer con las dos manos en el volante, por tal motivo se permite hacer uso del “manos libres”. [2]

### 1.3 Justificación

La importancia de contar con un equipo electrónico que permita adquirir los datos del vehículo radica principalmente en la seguridad del vehículo y del conductor, así como en el desempeño del carro durante la competencia.

Todo automóvil, sea éste de carreras o no, cuenta con un tablero que indica su estado mecánico y eléctrico, de esta manera es fácil, para quien maneja, saber si existe algún problema en alguna parte del vehículo o simplemente saber el consumo de combustible o la velocidad a la que se encuentra conduciendo.

Como el objetivo principal del equipo *Clean Mobilities* es hacer un buen papel en la competencia o mejor aún poder ganarla, el tener un sistema de adquisición de datos y de telemetría facilita la realización de esta meta.

Es importante que quien maneje el vehículo no sobrepase ciertos límites de velocidad por dos motivos: reglamento de *Shell* y para mayor eficiencia del carro. Además, por cuestión de seguridad debe conocer también si hay presencia de hidrógeno en el tanque, esto debido a que el carro funciona a base de una celda de hidrógeno. Cabe señalar que el proyecto no es innovador en cuanto a la idea de tener un tablero dentro del carro, pero sí lo es en cuanto a las tecnologías de comunicación (comunicación inalámbrica por *Wi-Fi*) que emplea y los dispositivos que se agregan (por ejemplo el *iPad* como tablero), en el capítulo dedicado al desarrollo del sistema se detallan dichos dispositivos, aplicaciones y tecnologías que dan solución a nuestra problemática.

Por otro lado, también es relevante que el resto del equipo que se ubique en los *pits* durante la carrera, tenga conocimiento de valores más específicos de la celda de hidrógeno y del motor, es aquí donde la telemetría juega su principal papel. La telemetría (medición a distancia), en nuestro caso, se hace por medio de almacenamiento de datos para que cuando el carro llegue a revisión a *pits* se pueda tener acceso a la información almacenada y facilitar el análisis de información para la toma de mejores decisiones que ayuden a obtener el primer lugar en la competencia con el mejor desempeño, eficiencia y el consumo mínimo de hidrógeno.

### 1.4 Objetivos

#### 1.4.1 Objetivo general

Desarrollar un sistema de adquisición de datos y de telemetría para un *kart*, con el fin de monitorear el estado del vehículo y tener un mejor desempeño durante la competencia.

#### 1.4.2 Objetivos Específicos

Para poder lograr el objetivo general, primero se debe definir un conjunto de objetivos específicos que nos permitan identificar de mejor manera los pasos a seguir para llegar a un resultado general.

A continuación se enumeran los objetivos específicos que consideramos son los más relevantes.

**a) Definir el equipo necesario para la adquisición de datos.**

En general casi en cualquier proyecto, una limitante fuerte es el presupuesto con el que cuenta el equipo, por tal motivo hemos tomado decisiones de material basadas en la disponibilidad de dicho equipo dentro del campus y, en general hemos tratado de dar resoluciones económicas a los diferentes problemas.

Básicamente, la principal preocupación de nuestro equipo en términos de elegir una tarjeta de adquisición de datos es la facilidad de aplicación y los beneficios que ofrece en términos de lo analógico y / o canales digitales para la lectura de datos, así como el tipo de protocolos de transmisión que utiliza.

**b) Definir el equipo a usar como panel de instrumentación**

Para esta cuestión, casi cualquier dispositivo de visualización puede funcionar, tales como una pantalla táctil RGB, PDA o incluso un teléfono celular puede ser utilizado, ya que sólo se encargaría de mostrar los datos en forma de un panel de instrumentación y no permitirá al piloto de *kart* tener ningún tipo de control de los parámetros mediante el panel, lo que significa que sólo servirá para mostrar la información de una manera estética que le da a nuestro *kart* una imagen profesional.

Nuestra opción inmediata es el uso de un iPad, un dispositivo muy reciente y de lujo, que podría ser proporcionado a nosotros por medio de un patrocinio, debido a que la institución tiene fuertes lazos con la Compañía Apple.

**c) Establecer un tipo de comunicación entre la tarjeta de adquisición de datos y el panel de instrumentación.**

Para este punto deseamos establecer un tipo de comunicación inalámbrica tomando en cuenta el dispositivo que se elija como panel de instrumentación. Esta comunicación entre la tarjeta de adquisición de datos y el panel de instrumentación es muy necesaria, debido a que el segundo estará poniendo al tanto al piloto de la información que se obtenga acerca del rendimiento del vehículo.

Debemos definir qué tecnología se adapta mejor a nuestras necesidades, en particular tendremos que decidir entre las dos más comunes, ya sea para redes de área local o redes de área personal: *Wi-Fi* y *bluetooth*. Ya que sólo necesita una conexión punto a punto en esta fase, el sentido común sugiere el uso de la tecnología *Bluetooth* debido a su

simplicidad en su aplicación y porque sería la opción más económica entre las dos opciones mencionadas anteriormente.

**d) Implementar la comunicación inalámbrica entre la tarjeta de adquisición de datos y el panel de instrumentación**

La comunicación inalámbrica entre los dos dispositivos se hará por medio de *Wi-Fi* (*Wireless Fidelity*), más adelante en el documento se explicará el funcionamiento de éste y el equipo necesario para lograr este tipo de conexión

**e) Definir el tipo de telemetría que usaremos**

En este punto es necesario analizar y definir si la telemetría se puede realizar en tiempo real u optar por una solución alterna.

**f) Almacenar datos en una tarjeta SD**

La tarjeta de adquisición de datos que ocuparemos cuenta con un puerto para la tarjeta SD, solo es cuestión de revisar cómo hacer la configuración del módulo destinado a la SD. Es importante especificar qué valores se guardarán en la SD ya que no serán los mismos datos que se envíen al panel de instrumentación que los que se guarden en la SD para leer posteriormente en *pits* por medio de una laptop. Cabe señalar que para poder programar en la EVK (tarjeta de adquisición de datos) es necesario retomar los conocimientos adquiridos en cursos anteriores para la programación sobre el AVR 32.

**g) Realizar una red inalámbrica en *Pits***

Para este punto es necesario indispensable conocer los distintos tipos de comunicación inalámbrica que se pueden establecer dentro del área de *pits*.

El objetivo de hacer una red inalámbrica en *pits*, es que se pueda compartir la información adquirida de cada parámetro a cualquiera de los integrantes que tengan algún dispositivo móvil que soporte la tecnología que se usará en el iPad.

**h) Desarrollar una interfaz gráfica agradable para los dispositivos móviles en *pits***

Se quiere tener una aplicación que despliegue los parámetros necesarios, estos parámetros tendrán un formato para desplegarse, sin embargo éste debe ser comprensible y amigable con el usuario, ya sea en forma de tablas o gráficas.

## 1.5 Alcances y limitaciones.

Los recursos económicos, humanos y de tiempo, la mayoría de las ocasiones, producen un obstáculo para la correcta culminación de los proyectos, en el nuestro no fue la excepción, por tal motivo en esta sección se explica hasta dónde abarca el proyecto, qué puntos se lograron completar y cuáles se podrían mejorar.

El sistema logra recibir datos digitales provenientes de un microcontrolador encargado del control de la celda de hidrógeno y del motor del vehículo; con los datos recibidos realizamos procesamiento y selección, es decir, unos se envían por *Wi-Fi* hacia el iPad (tablero del carro) y otros se almacenan en una tarjeta SD para su posterior lectura. Un alcance logrado fue la comunicación inalámbrica dentro del carro y el despliegue de la información, sin embargo una de las limitaciones que presenta el proyecto es en la telemetría.

Por cuestiones de tiempo no fue posible implementar la telemetría en tiempo real, ¿a qué me refiero con esto? La telemetría en tiempo real sería que el equipo ubicado en *pits* tuviera acceso al estado eléctrico y mecánico del carro, sobre todo a parámetros de la celda de hidrógeno y del motor, esto mientras el carro se encuentra en funcionamiento fuera de *pits* durante la competencia. Con la telemetría en tiempo real se podrían tomar decisiones al instante e informarle al piloto para tener un mejor desempeño.

En un principio se había pensado desarrollar la telemetría con el uso de tecnología 3G para mantener un envío constante de datos, desgraciadamente nos vimos limitados económicamente ya que se hubiera necesitado equipo adicional y algún plan tarifario conveniente para el constante envío de información. La manera de dar solución a la telemetría, aunque no fuera en tiempo real, fue por medio del almacenamiento de datos.

Para la lectura de datos es indispensable contar con una laptop con puerto SD integrado y/o un dispositivo móvil con *Wi-Fi* debido a que se crea una red inalámbrica local para que más de un dispositivo tenga acceso a los datos, en caso de que cada persona quiera visualizar información diferente en el mismo tiempo. Por ejemplo, si existe un grupo de 3 personas en *pits*, uno deberá crear la red en la laptop, los otros dos integrantes deben entrar a la dirección web indicada y así cada uno en su dispositivo móvil puede ver la información que más le convenga saber: uno podría ver el voltaje de la celda de hidrógeno, mientras que el otro la corriente del motor.

El sistema desarrollado puede ser instalado en cualquier vehículo, solo se harían los cambios pertinentes dependiendo el tipo de carro, para que despliegue sus datos característicos.

## Capítulo 2. Marco teórico

---

En esta sección primeramente se realiza un trabajo de investigación que nos permite conocer los sistemas parecidos al nuestro, con ayuda de la investigación podemos saber qué tecnologías se han empleado, qué es lo que podemos ocupar, qué mejoras podemos lograr en nuestro sistema y conocer las competencias que ya existen en el mercado.

### 2.1 Antecedentes

En este apartado enumeramos y analizamos las características de proyectos similares al que desarrollamos. Vemos las propuestas que otros equipos han dado para resolver la problemática que se les ha presentado para la adquisición de datos y de telemetría. Sobre todo nos enfocamos en los equipos que han participado ya en la carrera *Shell-Eco Marathon* o que planean participar.

Algunos de los equipos que han participado son los que se describen a continuación, junto con sus aportaciones para la adquisición de datos y de telemetría.

#### a) *Ecocustom Team*

*Ecocustom* comenzó su trayectoria alrededor del año 2007 con el nacimiento del primer diseño de *Urban Concept*, de la Universidad Politécnica de Valencia, campus de Alcoi.

En cuanto a la electrónica, específicamente a la parte a cargo de la adquisición de datos, el equipo utiliza un circuito impreso para el control de las funciones generales del vehículo. En dicho circuito, se incorpora la lectura analógica de nivel de combustible, baterías y supercondensadores tensiones, así como las lecturas digitales de RPM (revoluciones por minuto) y la velocidad. La medición de los niveles de combustible se realiza mediante un sensor óptico. Además utilizan un sistema de comunicación bluetooth con el fin de controlar ciertas funciones del vehículo (arrancar el motor o encender las luces) a través de cualquier dispositivo móvil, como PDA, ordenador portátil o un teléfono celular.

Finalmente implementaron un sistema de telemetría por medio de GPRS (*General Packet Radio Service*), que permite al equipo recibir los datos principales del vehículo, tales como el nivel de combustible, la presión y el posicionamiento GPS.

#### b) *Otros equipos participantes en Shell-Eco Marathon*

Existen otros equipos que han participado en la *Shell-Eco Marathon* en años anteriores, los cuales nos han servido de referencia para conocer qué tipo de tecnologías usan.

*Team Synergy* de la Universidad de Ingeniería y Tecnología de Lahore, Pakistán; el equipo *Cocodrilos* de la Universidad de Cambridge, Inglaterra y el equipo *Atomotive* de la

Universidad de Coventry Inglaterra no implementan ningún sistema de telemetría ni comunicaciones inalámbricas. La electrónica de estos equipos se basa en las baterías y luces de los *karts*, por lo tanto no cuentan con ningún tipo de control electrónico, sino pura mecánica. [4],[5][6]

**c) Escudería Borregos: Electrón**

La Escudería Borregos del Tecnológico de Monterrey Campus Ciudad de México ha participado continuamente en el Campeonato Electrón de Toyota LTH y ha recibido varios premios, incluyendo el primer lugar en la carrera en el 2010.

Todo el sistema se centra en el microcontrolador PIC18F4620, que controla el motor y lee los datos importantes, como la tensión de la batería, temperatura, porcentaje de aceleración, la velocidad de salida de corriente y, a continuación, los registra en una tarjeta SD. El equipo es capaz de recuperar la tarjeta SD del vehículo y conectarlo a un ordenador con el fin de analizar los datos utilizando Microsoft Excel

El equipo intentó implementar un sistema de telemetría, sin embargo éste no fue el más óptimo debido a que debían instalar antenas a lo largo y ancho de la pista de carreras y no usaba un esquema de modulación adecuado, por tal motivo usan almacenamiento de datos. [7]

**d) Spacelink<sup>NGT</sup>**

Los productos de *SpaceLink* son desarrollados por BV Servicios de Satélite en los Países Bajos. La compañía tiene más de 20 años de experiencia relacionada con sistemas de telecomunicaciones y telemetría, con un enfoque 100% en los datos. [8]

Ejemplo de productos de la familia *Spacelink<sup>NGT</sup>* incluyen:

- Simuladores de telemetría
- Gateways TM/TC (Telemetría/Telecomunicaciones)
- Transceptores de 70/230 MHz
- Procesadores de banda base TM/TC

**e) KMT Telemetry**

Es una empresa fundada en el año 1979 y ubicada en Munich, Alemania. Un año después entró al ámbito de los sistemas móviles de adquisición de datos y en el diseño del primer codificador PCM. EN el año 190 se concentra en la producción de equipos de telemetría. KMI tiene gran experiencia en el campo de las telecomunicaciones. Sus sistemas se basan en la adquisición de datos, acondicionamiento de señales y análisis de tensión, fuerza, desplazamiento, temperatura y vibración.

KMT cuenta con experiencia en líneas de producción para vehículos, aviones y coches de carreras. Entre sus productos se encuentra un sistema de telemetría punto a punto.

De acuerdo a la información obtenida, existen equipos de telemetría por satélite y probados, sin embargo no han sido aplicados en las competencias de autos organizadas por *Shell*. La idea de Electrón nos pareció bastante óptima en vista de las dificultades para realizar la telemetría en tiempo real y por el costo de los equipos. El proyecto no es innovador en cuanto al tipo de problema que se resuelve, pero sí por la manera de resolverla y por el uso de ciertas tecnologías y dispositivos empleados. [9]

## 2.2 Marco teórico

Ahora se presenta una explicación general del funcionamiento de ciertas tecnologías y dispositivos que ocupamos para poder llevar a cabo el proyecto.

### 2.2.1 Comunicación *Wi-Fi*

Es una tecnología cuyas siglas significan *Wireless Fidelity* (Fidelidad Inalámbrica en español), la característica principal del *Wi-Fi* es ser un medio de comunicación inalámbrico, es decir, que no requiere de cables y se basa en la implementación de ciertos protocolos establecidos. [10]

El funcionamiento del *Wi-Fi* consta de la propagación de ondas. Pertenecen a las redes WLAN (*Wireless Local Area Network*) y está regido por el estándar de la IEEE 802.11.

Existen dos tipos de comunicación *Wi-Fi* principales regidos por los siguientes sub-estándares:

- a) 802.11b. Que existe desde 1999 y es el más común, su ancho de banda de operación está alrededor de los 2.4 GHz con una velocidad máxima de 11 Mbps.
- b) 802.11g. Se estrenó en el año 2003, su ancho de banda de operación es también de 2.4 GHz pero con una velocidad máxima de 54 Mbps.

La banda de los 2.4 GHz incluye 14 canales de entre 20 y 22 MHz cada uno. Los dos tipos de comunicación *Wi-Fi* operan en esta banda y no es exclusiva del *Wi-Fi* también otras tecnologías como *bluetooth* trabajan a 2.4 GHz

#### Selección de la conexión inalámbrica

- Acceso extendido. La ausencia de cableado hace posible que el *Wi-Fi* llegue a zonas donde es imposible que los cables lleguen o que si llegan implique más costo.
- Reducción de costos. Además de que la ausencia de cables disminuye los costos, los enrutadores inalámbricos (necesarios para la conexión) son relativamente económicos y



no necesitan una inversión extra de excavación, perforación y otro método que sí requiere una conexión fija.

- **Movilidad.** Con esta comunicación inalámbrica se tiene la facilidad de desplazarse de un lugar a otro sin perder la conexión.
- **Multipataforma.** Actualmente casi todo dispositivo electrónico (teléfonos celulares, iPad's, iPod's, laptops, PC's, etc) tienen la opción de conectarse a una red *Wi-Fi*. [11]
- **Fácil conexión con iPad.** El iPad tiene ciertas restricciones para las comunicaciones de manera inalámbricas, sin embargo con *Wi-Fi* se puede establecer la conexión aunque ésta no sea exclusivamente entre dos dispositivos de Apple. [9]

### **Conceptos básicos de una red *Wi-Fi***

Todos los dispositivos móviles que se unen a una red *Wi-Fi* ya sea móvil, portátil o fija, se llaman estaciones inalámbricas (STA). Una estación móvil puede ser un PC, un ordenador portátil, una PDA, un teléfono o un módulo *Rabbit*. Cuando dos o más emisoras están conectadas de manera inalámbrica, forman un conjunto de servicios básicos (BSS).

Existen dos modos de operación del *Wi-Fi*: *ad-hoc* e infraestructura.

En el modo de operación *ad-hoc* las estaciones inalámbricas se comunican directamente entre sí siguiendo un modelo de red *peer-to peer*. En este tipo de modo no hay conexiones con otras redes *Wi-Fi* o con cualquier otra red LAN cableada. Debido a que este tipo de modo es fácil de implementar en cualquier área, se utiliza principalmente en situaciones que exigen una rápida instalación, por ejemplo en sitios de emergencia o en zonas de combate. *Ad-hoc* es una forma barata de tener conexión a Internet en un segundo equipo sin la necesidad de comprar un Punto de Acceso. Para este uso la configuración se dificulta ya que tenemos que configurar en el ordenador que tiene la conexión a Internet un programa enrutador o una conexión compartida.

El modo de infraestructura requiere que la BSS tenga un punto de acceso inalámbrico (AP por sus siglas en inglés de *Access Point*), el cual tiene como función ampliar el acceso a las redes de cable para los clientes de una red inalámbrica. En este modo de operación todos los móviles que traten de unirse a una BSS deben asociarse con un AP. El punto de acceso proporciona la entrada a las emisoras asociadas a lo que se le llama sistema de distribución (DS). [9]

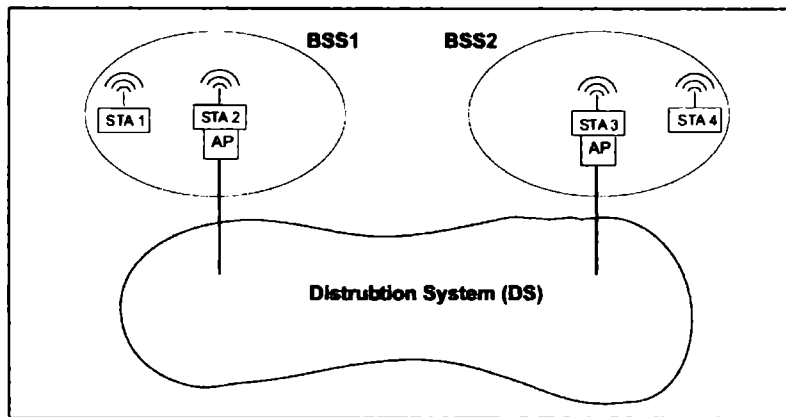


Imagen 1. Diagrama de una red por Wi-Fi [9]

Un Punto de Acceso es un dispositivo que crea un puente entre una red cableada una inalámbrica. Es como la antena a la que nos conectamos.



Imagen 2. Punto de Acceso

El protocolo 802.11 define a la AP como una estación STA que proporciona el acceso a los servicios de distribución (DS). Un enrutador inalámbrico también es compatible con el 802.11 ya que contiene la funcionalidad de AP. La diferencia principal entre un punto de acceso y un enrutador es que el segundo permite a los clientes tener acceso a redes de múltiples puntos de acceso, mientras que los AP solo permiten al cliente tener acceso a una sola red.

### Instalación de una red *Wi-Fi*

Para montar una red inalámbrica *Wi-Fi* se requiere de un Punto de Acceso (AP) el cual se conecta directamente a un módem, también es necesario un dispositivo *Wi-Fi* (en caso de que nuestro PDA no tenga *Wi-Fi*) conectado a nuestro aparato o PC por medio de tarjetas PCI (interconexión de componentes periféricos). Se recomienda tener el Punto de Acceso en un lugar alto para una mejor recepción y emisión. El funcionamiento de la red se basa en enrutadores *Wi-Fi* la configuración del enrutador permite controlar el acceso a la información que se transmite por el aire.

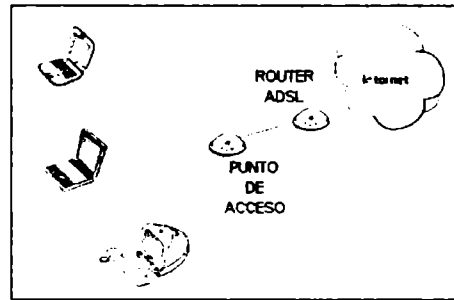


Imagen 3. Red inalámbrica *Wi-Fi* [9]

Para evitar la interceptación de la comunicación, es decir, que alguien no deseado tenga acceso a nuestra información se puede hacer uso de la encriptación de la transmisión, esto propicia una red más segura. También existe la opción de configurar el punto de acceso de manera que solo transmita a ciertos dispositivos. Para lograr esto último se usa la dirección MAC de los dispositivos que pueden entrar a la red. La dirección MAC (*Media Access Control address*) es un código único de identificación que tienen todas las tarjetas de red, los dispositivos con *Wi-Fi* tienen integrada esta dirección única y permanente preestablecida de fábrica.

Un punto de acceso o un enrutador inalámbrico que actúa como un punto de acceso, se suele configurar con un programa de utilidad proporcionado por el fabricante del dispositivo. Los clientes inalámbricos, tales como un dispositivo basado en un *rabbit*, se configuran a través de la aplicación de software que se ejecuta en el dispositivo inalámbrico.

Un *rabbit* es un dispositivo electrónico capaz de configurarse para recibir y/o transmitir por *Wi-Fi* en la sección destinada al diseño del software se detalla mejor la función y configuración de este aparato.

### 2.2.2 Comunicación Serial

Es importante dar una breve explicación del funcionamiento de la comunicación serial debido a que ésta es incorporada en nuestro sistema.

El RS-232 (Norma Recomendada 232, también conocida como *Electronic Industries Alliance RS-232C*) es una interfaz que especifica una norma para el intercambio de las secuencias de datos binarios entre un DTE (*Data Terminal Equipment*) y un DCE (*Data Communication Equipment*).

El RS-232 consiste en un conector DB-25 o DB-9 con un conjunto de señales de  $\pm 12V$ : +12 V para el estado lógico "0" y -12V para el estado lógico "1". [12]

Cada pin del conector tiene una función específica, en la siguiente tabla se muestran los pines principales y su función general

No.	Pin	Función
3	TxD	Transmisión de datos
2	RxD	Recepción de datos
4	DTR	Terminal de datos lista
6	DSR	Sistema de datos Listo
7	RTS	Solicitud de envío
8	CTS	Libre para envío

Tabla 1. Descripción de pines del serial

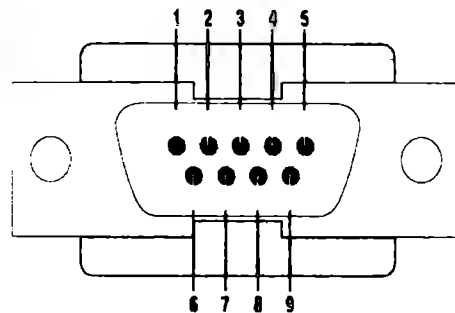


Imagen 4. Pines del puerto serial [13]

### Descripción de pines

**TxD.** La salida del serial RS232 es denominada línea de transmisión de datos. La señal en esta línea comprende una larga secuencia de pulsos generados por el puerto serial UART (Transmisor-Receptor Asíncrono Universal).

**RxD.** La secuencia de bits procedente de un puerto serial distante pasa a través de la línea de recepción de datos para llegar a la entrada de los puertos seriales del UART. La terminal de datos escucha en esta línea la señal de los datos procedentes del conjunto de datos.

**DTR.** Determina cuando la terminal de datos puede tomar parte en las comunicaciones, es decir, que se enciende cuando está listo para recibir o enviar otro dato.

**RTS.** Esta señal indica si el conjunto de datos puede ser enviado a la terminal de datos. La ausencia de una señal de estrategia en tiempo real a través de la conexión en serie impedirá el envío del conjunto de datos en serie. Este pin permite que la terminal de datos pueda controlar el flujo de datos.

La velocidad de transmisión más común es de 9600 bits por segundo y los datos pueden ser transmitidos por grupos de 5 a 8 bits. Una trama está formada por un bit de inicio, bits de datos, un bit de paridad y uno o dos bits de parada. El bit de paridad se utiliza para detectar errores. El formato más utilizado para enviar datos a través del puerto serie es el 8N1: 8 bits de datos, sin paridad y un bit de parada

### 2.2.3 EVK1100

Se trata de un kit de evaluación de Atmel que utiliza un microcontrolador AVR32 de la familia UC3A. Tiene muchos periféricos en la tarjeta, cuenta con un puerto Ethernet, dos puertos seriales, sensores de luz, sensores de temperatura, una pantalla LCD, conector JTAG, conector USB, interfaz de inmersión recíproca, interfaz SPI, SD y un lector de tarjetas MMC. [14]

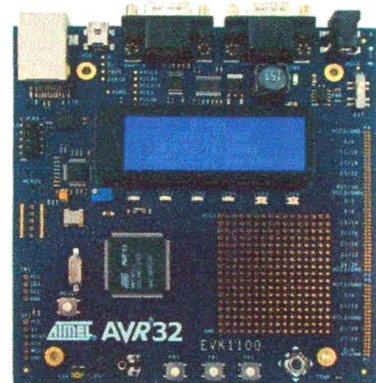


Imagen 5. EVK1100

La tarjeta EVK es la base del funcionamiento de nuestro sistema, en ella controlamos la adquisición de datos proveniente de un microcontrolador externo conectado por serial, hacemos análisis de información, mandamos datos hacia un panel de instrumentación y almacenamos en la tarjeta SD. Prácticamente toda la adquisición de datos y telemetría se encuentra programada en el microcontrolador de la tarjeta EVK.

El corazón de la tarjeta EVK1100 es el microcontrolador. El AT32UC3A0512 es un microcontrolador basado en el procesador RISC de AVR32, funciona a frecuencias de hasta 66 MHz. El AVR32 UC es el núcleo del microprocesador RISC de 32 bits de alto rendimiento, está diseñado para aplicaciones en sistemas embebidos sensible a los costos, con especial énfasis en el bajo consumo de energía, alta densidad de código y alto rendimiento.

#### Especificaciones del UC3A0512

- Microcontrolador AVR®32 UC 32-Bit de baja potencia y alto rendimiento
  - Conjunto de instrucciones RISC de ciclo sencillo, incluyendo conjunto de instrucciones DSP.
  - Modificación de lectura y escritura de las instrucciones y manipulación de bits
- Memoria flash interna de alta velocidad
  - Versiones de 512K Bytes, 256K Bytes, 128K Bytes
- Funciones del sistema
  - Alimentación y Reloj Manager incluyendo reloj interno RC y un oscilador de 32 kHz.
  - Dos osciladores multifunción y dos circuitos PLL (*Phase-Lock-Loop*)
- Frecuencias de CPU y USB independientes
  - *Watchdog Timer, Real-Time Clock Timer*
- Un controlador de Protocolo Serial Síncrono
  - Con soporte para I2S
  - *One Master/Slave Two-Wire Interface (TWI)*, 400kbit/s I2C-compatible
  - Un convertidor Análogo/Digital con 8 canales de 10 bits

- Otras especificaciones técnicas del UC3A512:
  - 5V de alimentación I/Os
  - Alimentación de 3.3V

Algunos módulos del microcontrolador son: *Power Manager (PM)*, *General-Purpose Input / Output (GPIO)*, *Interrupt Controller (INTC)*, *Timer/Counter (TC)*, *Universal Synchronous/Asynchronous Receiver/Transmitter (USART)*, *Analog-To-Digital-Converter (ADC)* y *Serial Peripheral Interface (SPI)*. A continuación se explicará en qué consiste cada módulo que usamos en el proyecto.

### **Power Manager (PM)**

Este módulo controla el oscilador y el circuito PLL del microcontrolador, también permite generar reloj y resetear el dispositivo.

Características:

- Soporta dos cristales (osciladores) dentro de un rango de 450 kHz a 16 MHz
- Soporta dos circuitos PLL entre 80 y 240 MHz
- Soporta osciladores de 32 kHz para generar contadores en tiempo real.

El módulo PM proporciona salidas para relojes genéricos, los cuales pueden ir conectados a pines de salida del microcontrolador multiplexadas por líneas de GPIO.

### **General- Purpose Input/Output Controller (GPIO)**

Este módulo maneja los pines de entrada y salida de propósito general del microcontrolador. Cada línea I/O se puede dedicar como de uso general o ser asignado a una función especial del chip. El GPIO controla todos los pines I / O del microcontrolador AVR32. Los pines se manejan como los puertos de 32 bits, cada puerto tiene un conjunto de registros de configuración.

El GPIO puede estar programado para generar una interrupción cuando detecta un cambio en la línea de entrada o salida.

### **Interrupt Controller (INTC)**

El INTC recoge solicitudes de interrupción de los periféricos, les otorga una prioridad y entrega una solicitud de interrupción. El INTC admite hasta 64 grupos de interrupciones. Cada grupo puede tener hasta 32 líneas de petición de interrupción, estas líneas se conectan a los periféricos. La arquitectura AVR32 soporta 4 niveles de prioridad para las interrupciones regulares, enmascarables y una interrupción no enmascarable (NMI).

### Timer/Counter (TC)

El temporizador / contador incluye tres canales de temporizador/contador de 16 bits cada uno. Cada canal puede ser programado para realizar mediciones de frecuencia independientes, conteo de eventos, medición de intervalos, generación de pulsos, temporización de retardo y modulación de anchura de pulso.

### Universal Synchronous/Asynchronous Receiver/Transmitter (USART)

Este modulo proporciona un puerto serial síncrono *full duplex* de enlace serie asíncrono. El formato de la trama de datos es fácilmente programable, es decir es muy flexible para realizar modificaciones en cuanto a su configuración de parámetros (longitud de los datos, la paridad, el número de bits de parada). El receptor implementa error de paridad, error de trama y la detección de desbordamiento de error.

La siguiente imagen muestra el diagrama a bloques del módulo USART del microcontrolador. El USART puede estar controlado por un canal de PDCA (*Peripheral Direct Memory Access Controller*) para transmitir y recibir datos. Cada dato es transmitido o recibido por el módulo de GPIO previamente explicado. El módulo USART puede estar configurado también para generar una interrupción en el procesador y la frecuencia del *baudrate* puede proveerla el módulo PM o por un pin de entrada.

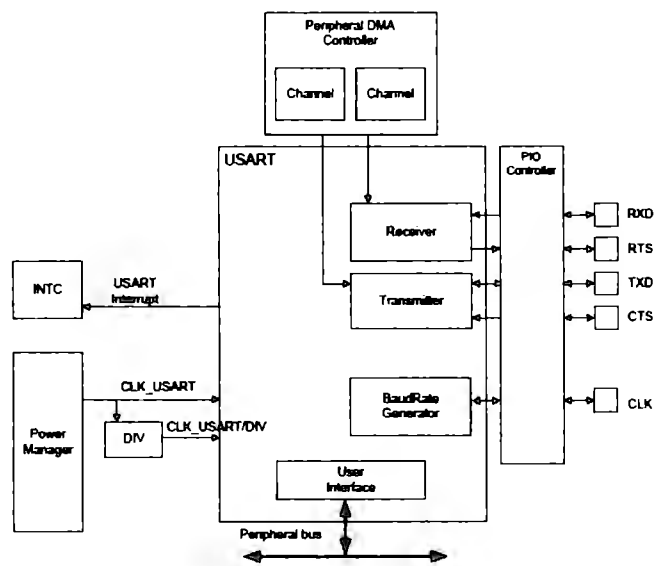


Imagen 6. Diagrama a Bloques de USART [14]

### **Analog-To-Digital Converter (ADC)**

Se basa en un convertido análogo/digital de 10 bits. Integra un multiplexor analógico de 8 entradas a una salida, lo que posibilita la conversión de 8 líneas analógicas, soporta conversiones de 8 bits o el modo de 10 bits de resolución.

#### Características

- Cuenta con Multiplexor integrado que ofrece hasta ocho entradas analógicas independientes e individuales
- Facilidad para activar y desactivar de cada canal
- Disparador por hardware o software
- Posibilidad de configuración de intervalos de ADC

### **SPI BUS**

Cuenta dos canales o líneas de datos de un bit. El SPI es compatible con sistemas MMC para el uso de tarjetas SD. La ventaja del modo SPI es que reduce el esfuerzo para su diseño. La desventaja del modo SPI es la pérdida de rendimiento en comparación con el modo de tarjetas SD. La EVK maneja las tarjetas SD en el modo SPI.

## 2.2.4 Tarjeta SD

Las tarjetas SD con memorias flash altamente integradas con capacidad de acceso aleatorio. Es accesible mediante una interfaz serial dedicada y optimizada para una rápida transmisión de datos. Esta interfaz permite a varias tarjetas ser apiladas a través de una conexión de sus periféricos.

El sistema de tarjeta SD es un nuevo sistema de almacenamiento masivo basado en las innovaciones en tecnología de semiconductores. Se ha desarrollado para proporcionar un medio de bajo costo, mecánica de almacenamiento robusta en forma de tarjeta para aplicaciones multimedia.

Un bajo consumo de energía y una amplia gama tensión de alimentación favorece a la aplicación en móviles y al consumo de baterías. La tarjeta SD proporciona la capacidad para todo tipo de datos multimedia usando sistemas eficaces de compresión de datos como MPEG.

La *SD Association* es una alianza mundial de más de 1.000 empresas que participan en el diseño, desarrollo, fabricación o venta de productos que utilizan la tecnología SD.

Las tarjetas SD están disponibles en tres formatos de capacidad: SD, SDHC y SDXC. Cada formato varía en clases de velocidad. [15]








	<b>Marcas</b>	<b>Operable bajo ...</b>	<b>Aplicaciones</b>	<b>Tarjeta de memoria SD</b>
Clase de velocidad		Normal I Bus / F	SD grabación de vídeo	SD, miniSD, microSD, SDHC, miniSDHC, microSDHC, SDXC, microSDXC
	 		- Full HD de alta definición de grabación de vídeo	
		De alta velocidad de bus de E / F	Full HD de grabación de vídeo HD sigue grabando consecutivos	
UHS clase de velocidad		SUS-I Bus / F	potencial más alto de la grabación de emisiones en tiempo real y captura de vídeos de alta definición de gran tamaño	SDHC UHS-I y SDXC UHS-I

Tabla 2. Características de los tipos de tarjetas SD [16]

La memoria de una tarjeta se divide en unidades de memoria mínimos. El host escribe datos en unidades de memoria en la que ya se dispone de datos almacenados. Como la memoria disponible se divide en unidades más pequeñas a través del uso normal, se conduce a un aumento en el almacenamiento no lineal, o fragmentada. El importe de la fragmentación puede reducir la velocidad de escritura, por lo que se necesitan tarjetas SD más veloces para poder compensar la fragmentación.

La tarjeta SD está dividida en sectores de 512 bytes cada uno. Una SD puede operar en dos modos: modo SD *Card* o modo SPI. Todas las tarjetas SD requieren soportar el SPI/MMC que incluye la interfaz de serie de cuatro cables (reloj, entrada serial, salida serial y selección de chip) compatible con los puertos SPI de muchos microcontroladores. El modo MMC no proporciona el acceso a las características propias de las tarjetas SD.

Existen 3 modos de transferencia para la SD:

- a) Modo SPI. Entrada separada serial y salida serial
- b) Modo un bit SD: Separa comandos, canales de datos y un formato propio de transferencia.
- c) Modo de cuatro bits SD: usa terminales extras para el soporte de transferencia paralela de cuatro bits.

Las tarjetas de baja velocidad transfieren de 0 a 400 Kbps en modo de un bit, mientras que las tarjetas de alta velocidad alcanzan una transferencia de 100 Mbps en el modo de cuatro bits y hasta 25 Mbps en modo de un bit.

### 2.2.5 RCM5400W

Es una tarjeta de desarrollo que contiene un microprocesador *rabbit* 5000. Proporciona un módulo compacto en un mini PCI con funcionalidad integrada para *Wi-Fi*/802.b/g que permite dar soluciones para este tipo de comunicación con bajo costo y bajo consumo de energía. [17]

Hemos decidido ocupar esta tarjeta como puente de comunicación entre la EVK1100 y el panel de instrumentación (iPad), debido a su facilidad de protocolos de comunicación que se pueden implementar en ella, como serial y *Wi-Fi*.

El RCM5400 tiene un conector USB que se comunica con la computadora, para poder programar el *rabbit* es necesario contar con *Dynamic C* versión 10.50, el cual incluye el protocolo *point-to-point* (PPP), sistema de archivos FAT y otras librerías.

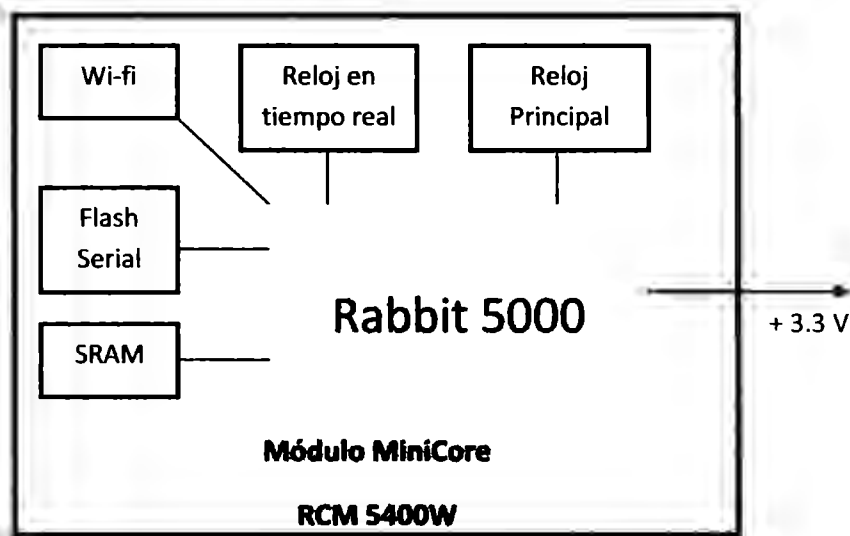


Imagen 7. Diagrama a Bloques de la tarjeta RCM5400W [17]

#### Características generales:

- Frecuencia de 73.73 MHz
- Conexión *Wi-Fi* 802.11b/g
- Memoria flash de 1 MB y SRAM de 512 kB
- 39 pines configurables de propósito general
- 6 puertos seriales

La transmisión del *Wi-Fi* es controlada por el chip *rabbit* 5000, el cual contiene Control de Acceso al Medio *Wi-Fi*. Como se mencionó anteriormente el *rabbit* implementa los estándares 802.11b/g.

El código del programa es almacenado en la memoria flash serial y se carga en la SRAM para ejecutarse en cuanto el módulo RCM5400W se alimenta. La interface de datos entre el procesador MAC y el *transceiver* AL2236 consiste en un convertidor D/A y otro A/D a 40 MHz.

El AL2236 es un chip *transceiver* que contiene receptor, transmisor, VCO, PLL y un amplificador de potencia para implementar 802.11b/g. La transmisión y recepción de datos se puede efectuar a más de 11 Mbits/s usando 802.11b y 54 Mbits/s usando 802.11g.[17]

Los canales *Wi-Fi* tienen un cierto grado de traslape entre sí. Cuanto más alejados dos números de los canales, menos la probabilidad de interferencia. Si se produce interferencia con una WLAN de vecinos, se debe hacer un cambio a un canal diferente. Por ejemplo, es conveniente usar los canales 1, 6 y 11 para reducir al máximo la interferencia.

### 2.2.6 iPad

Un iPad es un dispositivo electrónico desarrollado por Apple Inc. Funciona por medio de una NUI (Interfaz Natural de Usuario) sobre una adaptación del sistema operativo iOS. [18]

#### Características físicas:

Grosor : 1.32 cm  
Dimensiones: 24.28 x 17.87 cm  
Peso: 680 gramos

#### Especificaciones técnicas:

Pantalla con retroalimentación LED  
Memoria flash de 16 a 64 Gigabytes  
Bluetooth

Existen dos modelos de iPad, uno con conectividad a redes inalámbricas *Wi-Fi* 802.11n y otro con GPS y soporte de redes 3G (capaz de conectarse a redes de telefonía celular HSDPA).

En cuanto a los puertos, el iPad no cuenta con ranuras para tarjetas SD o para USB, sin embargo Apple ha desarrollado adaptadores para estos dispositivos con la finalidad de facilitar el intercambio de contenido multimedia con otros dispositivos con puertos USB o memorias SD (cámaras fotográficas, teléfonos móviles, etc.).

La función del iPad en el carro es ser un panel de instrumentación o tablero, que permitirá al conductor conocer de una manera gráfica y sencilla algunos parámetros relevantes del vehículo, como la velocidad

Se ha decidido ocupar un iPad por el gran impulso que existe hacia el desarrollo de aplicaciones para dispositivos móviles, el sistema que realizamos puede implementarse en cualquier vehículo, por lo tanto usar un iPad también brinda mayor comodidad al usuario. Existen diferentes dispositivos capaces de desempeñar la función de panel de instrumentación; una laptop por ejemplo, sin embargo sería demasiado estorbosa para el conductor del vehículo; otra opción era usar una pantalla LCD, pero los parámetros solo se desplegarían de manera digital, sin una interfaz gráfica amigable con el usuario. Así que por su diseño físico, ligereza, comodidad y por la capacidad que otorga al programador de realizar interfaces gráficas, el iPad fue una opción innovadora y ahora cuenta con una aplicación más.

## Capítulo 3. Desarrollo del Sistema

En esta sección se detallan las ideas y diseños que seguimos para realizar el proyecto y hacer funcionar todo en conjunto. El diseño del sistema involucra el diseño de hardware, diseño de software y sistema propuesto, se incluyen diversos diagramas y descripciones de cada dispositivo empleado, todo esto para facilitar la comprensión del proyecto.

### 3.1 Descripción general del sistema

En esta parte se detalla la propuesta del sistema por medio de diagramas a bloques y la explicación pertinente a cada diagrama para una mejor comprensión. Para hacer la propuesta del sistema fue necesario basarnos en la problemática y en la investigación previamente hecha del estado del arte para poder llegar a la solución más óptima y segura.

A continuación se presenta un diagrama a grandes rasgos de todo el sistema.

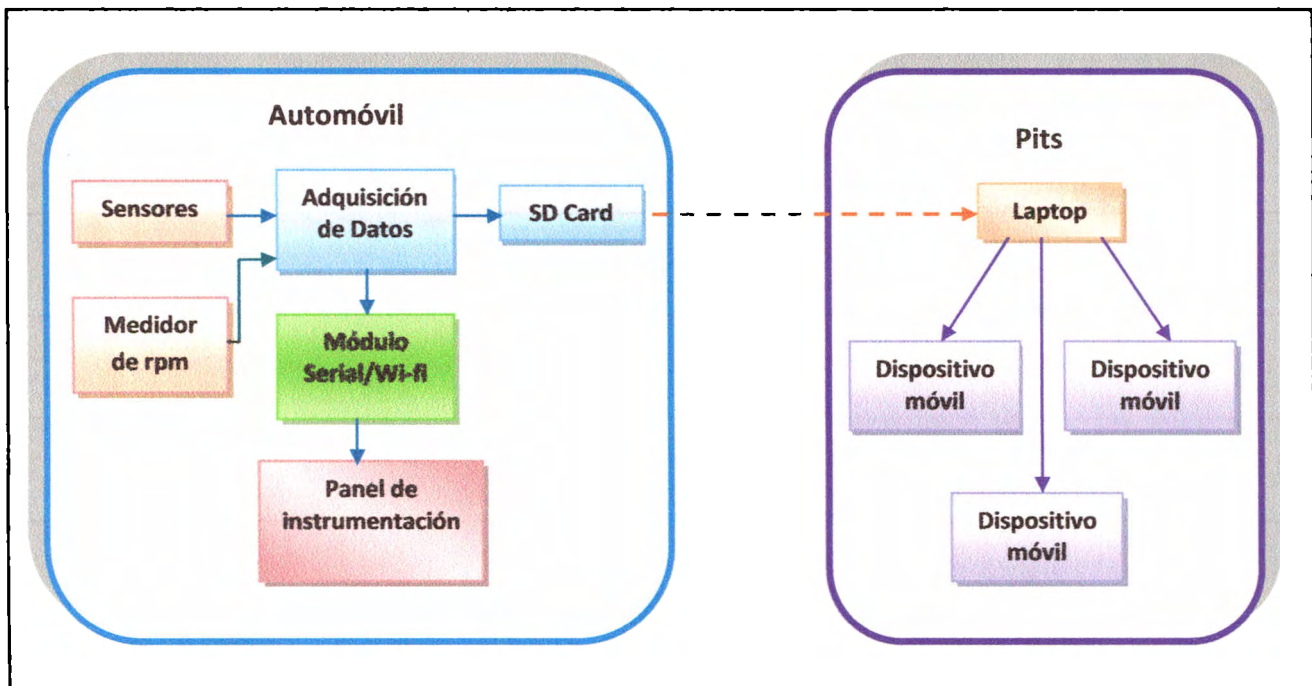


Imagen 8. Diagrama a Bloques del Sistema

Existen diversos parámetros referentes a la celda de hidrógeno y al motor que son necesarios conocer, por cuestiones de eficiencia, pero sobre todo por seguridad del vehículo y del conductor.

Otro equipo perteneciente a *Clean Mobilities* está encargado de medir los valores analógicos entregados por sensores de la celda de hidrógeno y del motor, una vez que tienen los valores los recuperan por medio de un microcontrolador y los digitalizan para enviarlos por un puerto serial del mismo. Nosotros adquirimos los datos de los sensores mediante un puerto serial de la tarjeta de adquisición de datos. Ya que tenemos todos los valores, seleccionamos cuáles de estos se envían al panel de instrumentación situado dentro del carro y cuáles se almacenan en la tarjeta SD para posteriormente leerlos en *pits*.

Los parámetros que se desplegarán en el panel de instrumentación deben ser enviados por otro puerto serial de la tarjeta de adquisición de datos hacia un módulo serial/*Wi-Fi* (RCM 5400W), este módulo previamente configurado de la manera adecuada se encarga de recibir los datos por serial y transmitirlos por *Wi-Fi*. Una vez que se han radiado los datos por *Wi-Fi* el panel de instrumentación (con *Wi-Fi* integrado) intercepta los datos. El panel de instrumentación elegido para desempeñar esta función es un iPad. Además de desplegar valores en el iPad, se implementó un sistema de alarmas; el sistema de alarmas está programado en el microcontrolador de la tarjeta de adquisición de datos y se despliega en el panel de instrumentación del carro. Una alarma es generada cuando se está fuera de los rangos de seguridad y eficiencia especificados para cada parámetro.

Los otros valores que no se despliegan en el panel son almacenados en una tarjeta SD que se ubica en la tarjeta de adquisición de datos. Durante la carrera, el vehículo debe hacer ciertas paradas en *pits*, cuando esto sucede, los integrantes ubicados en *pits* deben sacar la SD de la tarjeta e introducirla en una laptop, la cual leerá el contenido de la SD y funcionará como servidor de una red inalámbrica para que los demás integrantes del equipo ubicados en *pits* puedan tener acceso por medio de dispositivos móviles (iPad, iPod, Blackberry, etc.) a la información leída en la laptop.

Como se puede observar no existe una telemetría como tal en tiempo real, lo que hacemos para sustituirla es grabar en un dispositivo de almacenamiento la información relevante del vehículo para leerla en *pits* y poder desempeñar un mejor papel en la carrera.

## 3.2 Diseño del prototipo

El desarrollo de todo el sistema se vio dividido en dos etapas: etapa de adquisición de datos y etapa de telemetría.

### 3.2.1 Sistema de adquisición de datos

Se explicará en esta parte de manera más específica en qué consiste nuestra propuesta para el sistema de adquisición de datos, qué parámetros recibimos, por qué y cómo se envían hacia el panel de instrumentación. Principalmente los datos que el conductor del auto necesita observar en su tablero son la velocidad a la que se encuentra manejando, el flujo de hidrógeno y alarmas en caso de que algún elemento no esté funcionando de manera adecuada en el sistema.

El *hardware* empleado para realizar la etapa de adquisición de datos es el siguiente:

- a) Circuito de acoplamiento
- b) Circuito medidor de revoluciones por minuto
- c) Tarjeta EVK1100 con microcontrolador UC3A512
- d) Rabbit TCM5400W (RS232/Wi-Fi)
- e) iPad

Las características y especificaciones técnicas de cada uno de los componentes están detalladas en el sustento teórico del capítulo 2 de este documento.

El proyecto estuvo basado en programación, cada componente se programa en lenguajes y plataformas diferentes, los únicos circuitos diseñados y construidos por nosotros fueron los que desempeñan las tareas de acoplamiento de señales y medición de revoluciones por minuto de la llanta.

El siguiente diagrama muestra todos los módulos y dispositivos usados para la etapa de adquisición de datos.

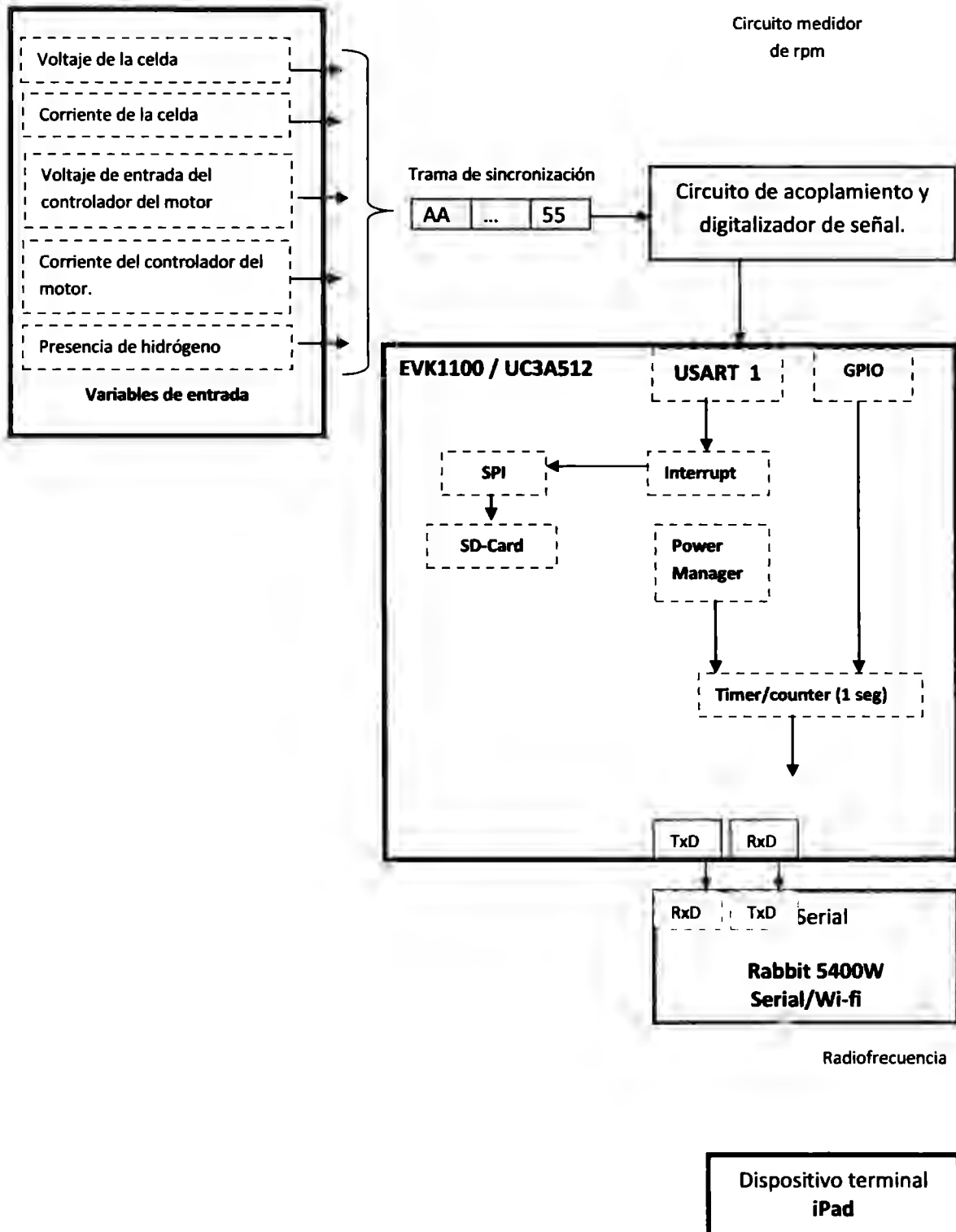


Imagen 9. Diagrama a Bloques del Sistema de Adquisición de Datos

Debido a que se está trabajando con sistemas independientes, es decir, un equipo está dedicado al control (celda y motor) y otro equipo nos encargamos del despliegue de la información, es necesario definir una trama que haga posible la sincronía entre los microcontroladores involucrados en las dos etapas.

En el diagrama a bloques de la imagen 9 se observa una trama de sincronización, la función de esta trama es asegurar la sincronía para poder recibir los datos de manera correcta y en caso de que no se reciban adecuadamente desecharlos.

La trama de sincronía tiene la siguiente estructura:

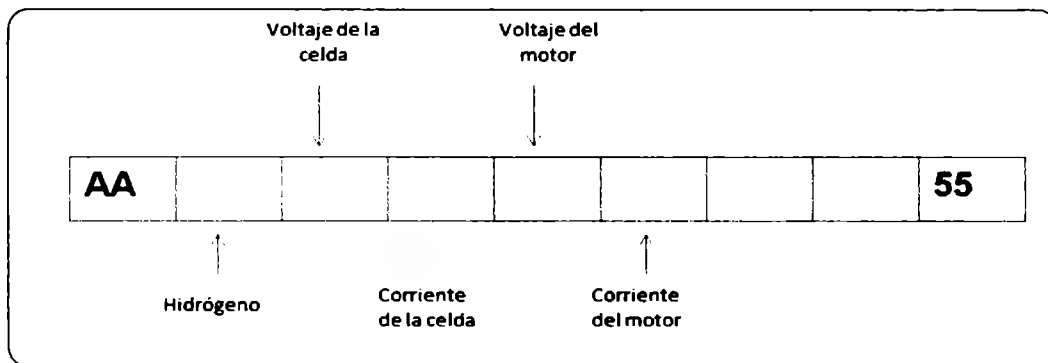


Imagen 10 Trama Sincronía

Cada espacio consta de un byte, el primero es un identificador de inicio de trama, el cual se estableció como "AA", el dato que continua es el valor correspondiente a la presencia de hidrógeno, seguido por el voltaje de la celda, la corriente de la celda, voltaje del motor y corriente del motor, finalmente, en el último espacio de la trama se envía un "55", el cual indica el fin de la trama.

En el programa del microcontrolador se detecta esta trama, en caso de que se pierda la sincronía, no se detectará correctamente cada dato de la trama, ni el 55 del final, por lo tanto se desecha esa trama y se conserva el valor anterior hasta recuperar la sincronía. La salida de la etapa de control con la trama de sincronía va directamente conectada al puerto serial USART 1 con un *baudrate* de 2400 bits/seg.

De todos los parámetros que se reciben de la etapa de control y del circuito medidor de revoluciones por minuto, los únicos datos que se despliegan en el tablero del vehículo son la presencia de hidrógeno y la velocidad.

### Presencia de hidrógeno

Este parámetro nos da información sobre el hidrógeno contenido en el tanque. Tenemos que saber la presencia de hidrógeno para calcular la presión de hidrógeno él. Sobre todo este valor se



requiere conocer por cuestiones de seguridad, debido a que el hidrógeno es un elemento altamente inflamable, por lo tanto peligroso.

En caso de no exista hidrógeno suficiente en el tanque, se enviará un aviso al iPad para que el conductor del auto esté informado y pueda tomar las decisiones pertinentes.

El valor del flujo de hidrógeno es recibido por medio del puerto serial 1 del microcontrolador de la EVK1100.

### Velocidad

Para cualquier conductor es de suma importancia observar la velocidad a la que se está desplazando el vehículo. Para medir la velocidad hemos desarrollado un circuito simple que permite contar los rpm (revoluciones por minuto). A continuación se presenta un diagrama a bloques de la medición de los rpm.

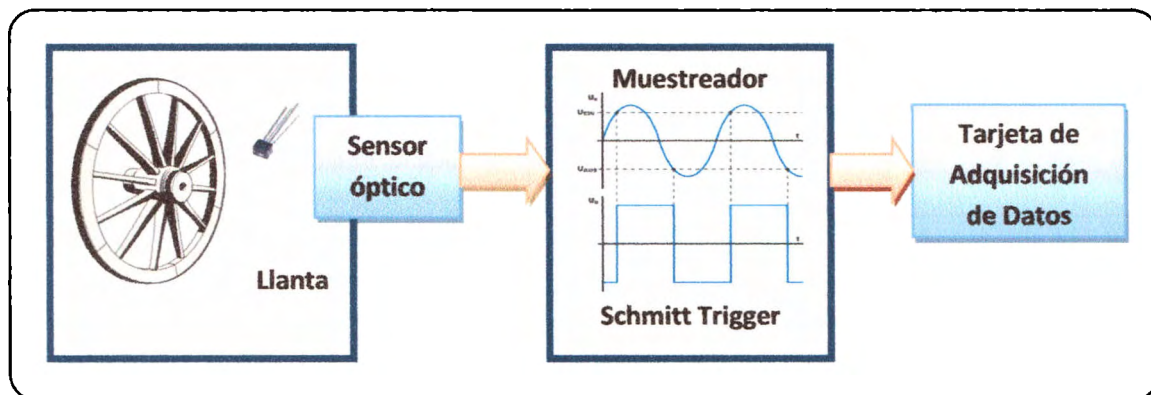


Imagen 11. Diagrama a Bloques de la medición de los rpm.  
Rueda[19], sensor [20], Schmitt Triger[21]

La manera de medir la velocidad es por medio de las revoluciones por minuto, para esto se colocó un sensor a un lado de la llanta.

El sensor escogido es un QRD1114, el cual funciona por medio de un diodo emisor infrarrojo y un fototransistor infrarrojo también para detectar las señales. Se aplica para detectar transiciones de negro y blanco o para la detección de objetos cercanos. El sensor entrega un voltaje diferente dependiendo la proximidad de los objetos, este caso detecta las ranuras que posee la llanta y entrega un nivel de voltaje de acuerdo a la detección que hace.

Debido al giro de la llanta la forma de onda que se forma es tipo sinusoidal, por lo tanto el siguiente paso es digitalizar esa señal. Para digitalizar la señal se diseñó un circuito basado en un *Schmitt Trigger*.

Un *Schmitt Trigger* es un circuito comparador con dos niveles de umbral diferentes de tensión que genera una señal de pulsos a la salida, fue inventado por Otto Herbert Schmitt.

El *Schmitt trigger* cuenta con una entrada y una salida, en la salida lo que se tiene es un cambio agudo y de manera muy rápida cuando el nivel de la señal de entrada excede un valor predeterminado. Por tal motivo este circuito es aplicado para transformar señales sinusoidales en cuadradas, para acondicionamiento eléctrico o para restaurar pulsos que han sido deteriorados por interferencias durante la transmisión. [22]

La siguiente imagen corresponde al circuito diseñado para muestrear la señal.

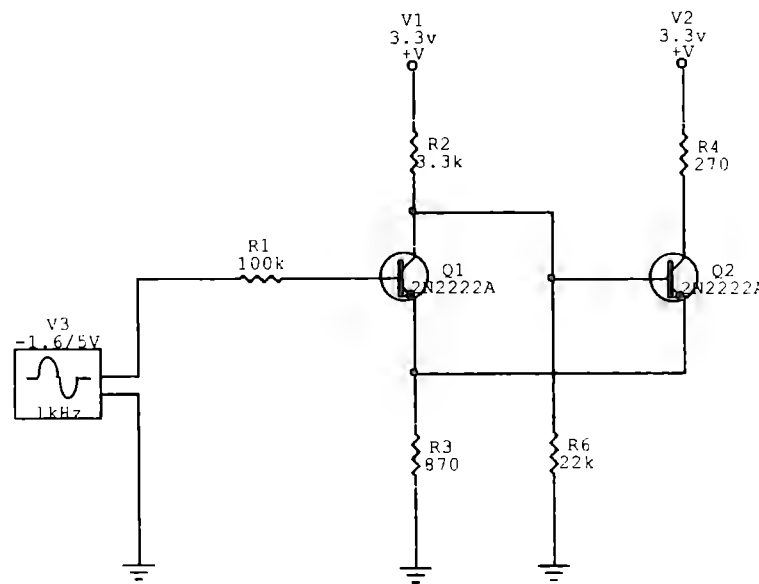


Imagen 12. Circuito muestreador de señal

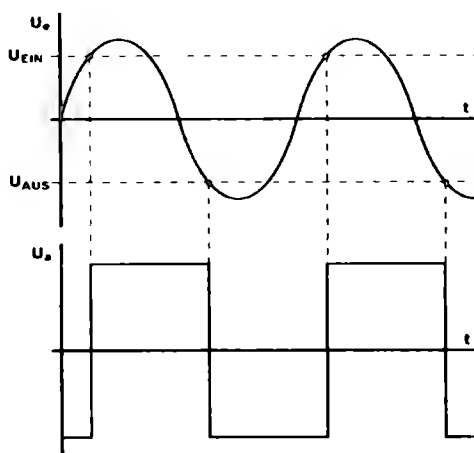


Imagen 13. Respuesta del *Schmitt Trigger* [21]

La tensión alta o baja es en realidad el voltaje de alimentación positivo o negativo que alimenta al comparador.

En la imagen 13 la primera gráfica corresponde a la entrada de corriente alterna. La línea punteada  $V_{EIN}$  indica el nivel alto del umbral y punteada  $V_{AUS}$  indica el nivel bajo del umbral. La segunda gráfica corresponde a la salida del *Schmitt Trigger*. Si el voltaje de la entrada está por encima del nivel alto de umbral la salida se encuentra en alto y si el voltaje de la señal de entrada está por debajo del nivel de umbral inferior la salida es baja.

El ancho del pulso depende de la velocidad con la que se mueva la llanta, si ésta gira rápidamente la respuesta tendrá una forma cuadrada, con un ancho de pulso muy pequeño. Por el contrario, si gira lentamente, el ancho del pulso será mayor.

La salida del circuito está directamente conectada a un pin de propósito general del microcontrolador UC3A512 de la EVK1100. Cuando ya tenemos el valor de los rpm's es necesario hacer una conversión en el microcontrolador para tener como resultado el valor de la velocidad en unidades de km/h.

La fórmula que nos permite obtener la velocidad es:

$$velocidad = 2\pi(\text{radio de la llanta})(rpm) \times \frac{1 \text{ km}}{1000m} \times \frac{3600 \text{ s}}{1 \text{ min}}$$

#### Medidas de la llanta

Parámetro	Valor
Diámetro	50 cm
No. de ranuras	16

Tabla 3. Medidas de la llanta

#### EVK1100/UC3A0512

El módulo PM (*Power Manager*) es el que le proporciona el reloj a todo el sistema, la frecuencia de oscilación es de 12MHz, con esta frecuencia se configura el *timer/counter* para que genere una interrupción en el sistema cada segundo de tiempo. Por lo tanto, al generarse la interrupción de que ha pasado un segundo lo que se hace es leer el valor recibido por USART 1 para el flujo de hidrógeno y almacenarlo en una variable, también lee el valor correspondiente a los rpm's, realiza la conversión necesaria y finalmente envía los valores a otro puerto serial del microcontrolador (USART 0).

#### Rabbit 5400W

El USART 0 está directamente conectado al *rabbit 5400W* y opera con un *baudrate* de 38,400 bits/seg. El *rabbit* está configurado para recibir datos mediante uno de sus puertos seriales y hacer la transmisión vía *Wi-Fi* (en el capítulo de diseño de software se especifica dicha configuración). Prácticamente el *rabbit* funciona como un puente entre la tarjeta de adquisición de datos y el iPad para poder hacer la transferencia de datos con protocolos diferentes.

#### iPad

Finalmente el iPad recibe la información por Wi-Fi y la despliega con una interfaz gráfica amigable con el usuario. Como se explicó anteriormente, el iPad sólo cumple la función de ser un tablero para el conductor.

La apariencia de la interfaz gráfica para mostrar los parámetros que recibe es como la siguiente.



Imagen 14. Interfaz gráfica del iPad (tablero)

### 3.2.2 Sistema de alarmas

Se implementó un sistema de alarmas puntuales para procurar el monitoreo del buen funcionamiento del vehículo, particularmente de las características eléctricas del mismo. Se les llama alarmas puntuales debido a que su disparo sólo depende de la variable que están monitoreando.

Estas alarmas únicamente sirven para verificar que los valores máximos permitidos por las hojas de especificaciones de los componentes nunca se sobrepasen. Fueron implementadas en el microcontrolador de la EVK de tal forma que se necesitan modificar unas constantes si se desea cambiar el valor máximo monitoreado por las mismas.

La siguiente tabla muestra los valores máximos, si algún dato sobre pasa este valor se produce una alarma en el sistema.

Parámetro	Valor máximo
Hidrógeno	1
Voltaje de celda	40 V
Corriente de la celda	40 A
Voltaje del motor	60 V
Corriente del motor	20 A

Tabla 4 Valores máximos de parámetros

Los valores fueron proporcionados por el equipo encargado de la etapa de control de acuerdo a las hojas de especificaciones leídas por ellos para la celda de hidrógeno y el motor.

Así mismo, se pensó en la implementación de otro sistema de alarmas, el cual no sólo tome en cuenta una sola variable para dispararse. Este segundo sistema consiste en monitorear continuamente las revoluciones por minuto del motor, así como la potencia entregada por éste.

Para diseñar este sistema se hace uso de lógica difusa sencilla, con método horizontal para su defusificación. Nuevamente, se debe buscar implementar esta lógica de manera generalizada dentro del microcontrolador para que equipos posteriores puedan hacer uso de la misma y sea sencillo modificar los valores de monitoreo. Las gráficas que representan el sistema difuso se muestran a continuación:

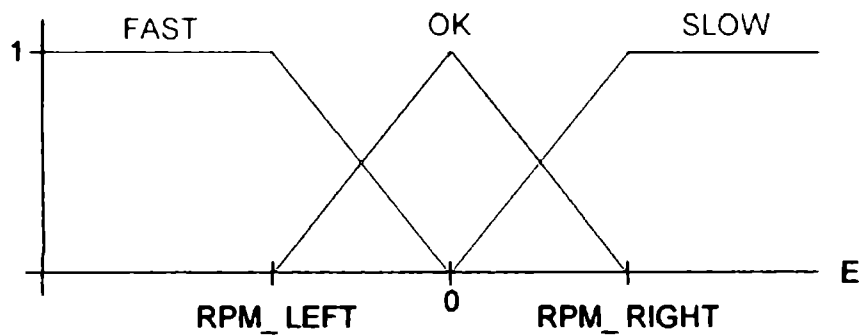


Imagen 15. Espacio difuso para los rpm del motor.

En la gráfica E representa el error calculado como la diferencia entre el valor deseado de las revoluciones por minuto y el valor sentido.

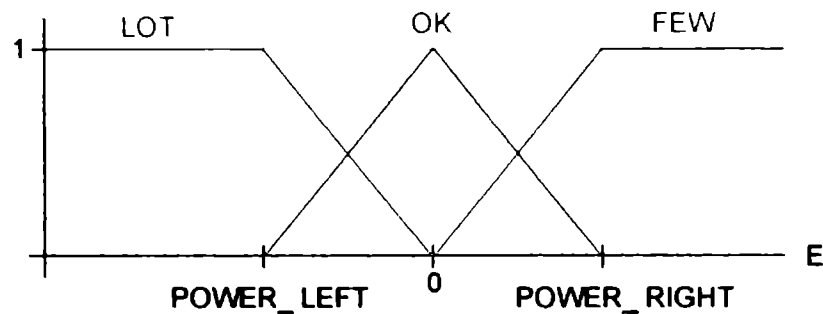


Imagen 16. Espacio difuso para la potencia entregada por el motor.

En la gráfica anterior E representa el error calculado como la diferencia entre el valor deseado de potencia entregada por el motor y el valor calculado a partir del voltaje y corriente sensadas.

En la teoría, los sistemas difusos siempre tienen un valor máximo de 1, pero en este caso nuestro valor máximo será 255 para poder implementar la lógica difusa en un microcontrolador de un mínimo de 8 bits; por lo cual el eje del error deberá tomar valores entre -128 y +127.

La gráfica del espacio de defusificación se muestra a continuación:

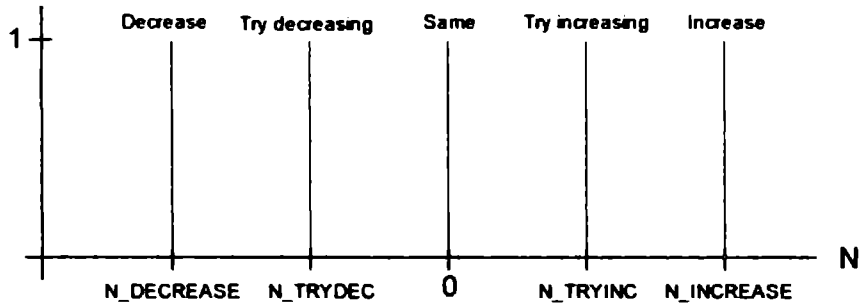


Imagen 17. Gráfica de defusificación

Donde la ubicación del punto a lo largo del eje N se calcula con la siguiente fórmula:

$$\Delta N = \frac{Dec * N_{DECREASE} + Trydec * N_{TRYDEC} + Tryinc * N_{TRYINC} + Inc * N_{INCREASE}}{Dec + Trydec + Same + Tryinc + Inc}$$

Donde Dec, Trydec, Same, Tryinc e Inc son las magnitudes en las cuales el valor sentido pertenece a las funciones de pertenencia propiamente, lo cual es calculado por la lógica difusa.

A partir del valor de  $\Delta N$  se puede emitir la alarma correspondiente de entre las posibles alarmas (mostradas en la gráfica anterior) dependiendo a qué alarma tienda más el valor de  $\Delta N$ .

Las reglas del sistema difuso se resumen en la siguiente matriz:

	FEW POWER	POWER OK	LOT OF POWER
SLOW	Increase	Try increasing	
RPM OK	Try increasing	Same	Decrease
FAST		Try decreasing	Decrease

Tabla 5. Tablas de reglas del sistema difuso

Lo cual genera las siguientes relaciones booleanas:

$$\begin{aligned}
 Increase &= SLOW \wedge FEW \\
 Try\ Increasing &= (SLOW \wedge OKpower) \vee (OKrpm \wedge FEW) \\
 Same &= OKrpm \wedge OKpower \\
 Try\ Decreasing &= FAST \wedge OKpower \\
 Decrease &= (OKrpm \wedge LOT) \vee (FAST \wedge LOT)
 \end{aligned}$$

En lógica difusa, las operaciones AND se realizan tomando el mínimo de los operandos, mientras que las operaciones OR se realizan tomando el máximo de los operandos.

Se decidió programar esta lógica en Matlab antes de implementarla en el microcontrolador para poder realizar pruebas y comprobar el buen funcionamiento de la misma.

Para realizar las pruebas en Matlab se tomó como referencia un valor deseado de las revoluciones por minuto de 320 y un valor deseado de potencia igual a 940 W. Estos valores fueron obtenidos de las curvas de eficiencia correspondientes al motor utilizado por el equipo de *Clean Mobilities*; con los cuales se logra el punto de mayor eficiencia, teóricamente alcanzable, en la práctica.

Así mismo, basándonos en dichas curvas de eficiencia, escogimos un rango de operación sobre el cual realizaríamos las pruebas. La siguiente tabla resume los valores elegidos:

Rango de operación permitido	Rango de operación de prueba
$220 \leq \text{RPM} \leq 340$ [rpm]	$0 \leq \text{RPM} \leq 425$ [rpm]
$840 \leq \text{POTENCIA} \leq 1120$ [W]	$0 \leq \text{POTENCIA} \leq 1500$ [W]

Tabla 6. Rangos de operación para rpm y potencia

El máximo del rango de operación de prueba para las revoluciones por minuto fue elegido considerando que el resto del equipo de *Clean Mobilities* estimó una velocidad máxima de 40 km/h para el vehículo. El máximo rango de operación de prueba para la potencia entregada por el motor se eligió al tomar los máximos de voltaje y corriente establecidos por la hoja de especificaciones del motor, menos 10 unidades cada uno.

Para el espacio de defusificación se eligieron los siguientes valores, considerando que el eje N sólo puede ir de -128 a +127 por las razones previamente explicadas:

Nombre del valor en el eje N	Valor asignado
N_DECREASE	-88
N_TRYDEC	-48
N_SAME	0
N_TRYINC	88
N_INCREASE	127

Tabla 7. Valores para la defusificación

Lamentablemente no es posible proporcionar un sustento para la elección de los valores anteriores, puesto que no hay nada establecido para ello. La elección debe hacerse, propiamente, tras ejecutar pruebas de campo, es decir, con el equipo real en condiciones reales.

Para realizar las pruebas se programaron 4 funciones:

Nombre de la función	Descripción
<b>Fuzzy</b>	Se encarga de hacer todos los cálculos para fusificar, conocer grados de pertenencia y defusificar. <b>Entrega el valor de <math>\Delta N</math>. Recibe el valor del RPM sensado y la potencia calculada.</b>
<b>FuzzyRPM</b>	Realiza un barrido de RPMs a través de llamadas recursivas a la función Fuzzy, antes descrita, sin variar la potencia. <b>Entrega un arreglo con los <math>\Delta N</math> calculados. Recibe un arreglo con las diferentes RPMs.</b>
<b>FuzzyPOWER</b>	Realiza un barrido de POTENCIAS a través de llamadas recursivas a la función Fuzzy, antes descrita, sin variar los rpms. <b>Entrega un arreglo con los <math>\Delta N</math> calculados. Recibe un arreglo con las diferentes POTENCIAS.</b>
<b>FuzzyBOTH</b>	Realiza un barrido tanto de RPMs como de POTENCIAS a través de llamadas recursivas a la función Fuzzy, tomando el valor del mismo índice de ambos arreglos. <b>Entrega un arreglo con los <math>\Delta N</math> calculados. Recibe un arreglo con los diferentes RPMs y otro arreglo con las diferentes POTENCIAS.</b>

Tabla 8. Descripción de las funciones del sistema difuso

Cabe señalar que el sistema de alarmas difusas no está implementado en la tarjeta de adquisición de datos, pero ya está la teoría realizada para los rpm y la potencia. Se realizaron simulaciones en Matlab para evaluar el funcionamiento del sistema de control difuso para las alarmas. El código y las gráficas resultantes de la simulación se encuentran en los anexos de este documento escrito.

### 3.2.3 Telemetría

Ahora se explicará el procedimiento usado para la telemetría. En la imagen 14 se observan los parámetros que se reciben por el puerto serial del microcontrolador con la intención de almacenarse en una tarjeta SD.

Las variables que los integrantes del equipo deben observar en *pits* usando la tarjeta SD son:

- a) Presencia de hidrógeno
- b) Voltaje de salida de la celda de combustible de hidrógeno
- c) Corriente de salida de la celda de combustible de hidrógeno
- d) Voltaje del motor
- e) Corriente del motor
- f) Potencia de la celda de combustible de hidrógeno
- g) Velocidad



### **Presencia de hidrógeno**

El objetivo de conocer la presencia de hidrógeno es el mismo que el que se tiene en la adquisición de datos, por cuestiones de seguridad principalmente. Además es importante medir este parámetro para poder hacer un análisis de la eficiencia del carro. El objetivo de la carrera no es precisamente ser el primero en llegar a la meta, sino que gana el vehículo que logre recorrer mayor distancia con el menor consumo de combustible.

### **Voltaje de salida de la celda de combustible de hidrógeno**

La celda de combustible de hidrógeno opera a cierto nivel de voltaje al igual que su controlador. El correcto funcionamiento de todo el sistema (es decir, del vehículo) depende de la correcta operación el controlador de la celda de combustible. Por lo tanto la medición del voltaje de salida de la celda es muy importante y debe ser leído por la gente ubicada en *pits*.

### **Corriente de salida de la celda de combustible de hidrógeno**

El valor de la corriente de la celda debe mantenerse en un rango determinado de acuerdo a la hoja de especificaciones de dicha celda. También el equipo ubicado en *pits* debe ser capaz de conocer este valor para poder prever posibles problemas y para dar un mejor apoyo de asesoría al conductor del vehículo.

### **Voltaje del motor**

El controlador del motor opera a un voltaje promedio equivalente a los 48 V, es por esto que requerimos conocer el valor que se entrega para poder asegurar que la tensión no sobrepase dicho valor y así asegurar el correcto funcionamiento del controlador.

### **Corriente del motor**

Al igual que el parámetro anterior, el controlador del motor opera a una corriente determinada, esto para que el sistema pueda ser seguro y eficiente, por lo tanto se debe asegurar que el valor de la corriente se encuentra en un rango aceptable, de lo contrario se debe informar al conductor del vehículo para que tome ciertas medidas y no ponga en riesgo su bienestar y el correcto funcionamiento del vehículo.

Como se observa en la imagen 18 todos los valores previamente mencionados son recibidos por uno de los puertos seriales del microcontrolador ubicado en la EVK1100. Cada que se reciben datos en el microcontrolador se genera una interrupción en el sistema, durante esta interrupción lo que se hace es leer los valores recibidos por serial, guardarlos en variables y posteriormente escribirlos en la tarjeta SD en caso de que ésta esté presente. Si el sistema no detecta la tarjeta SD prácticamente el sistema no funcionaría, es decir, esperaría hasta que la tarjeta fuera introducida.

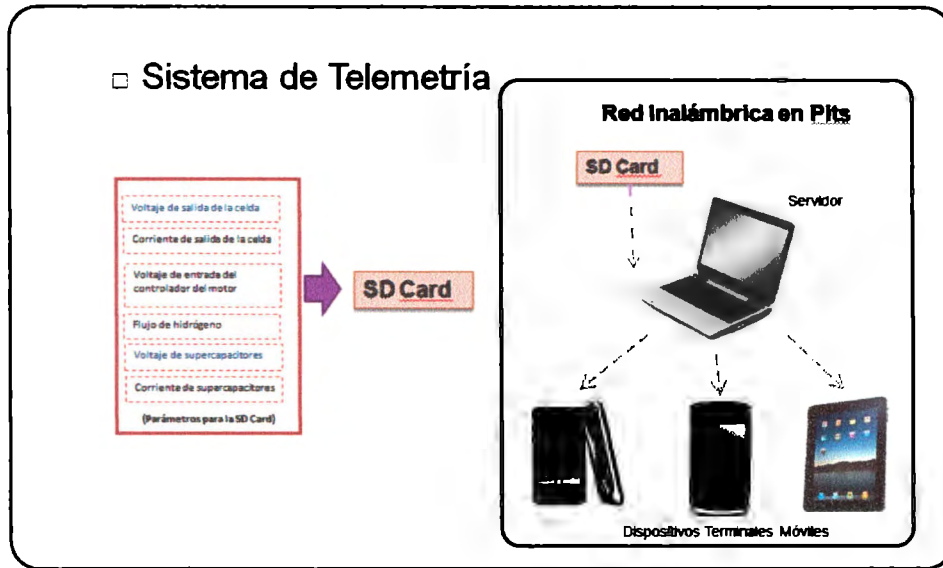


Imagen 18. Diagrama a bloques del sistema de telemetría

Es importante mencionar la manera en la que se almacenan los datos en la tarjeta SD. La tarjeta está dividida en una cantidad muy grande de sectores, lo que hacemos es otorgarle a cada parámetro muchos sectores continuos en los que se puede escribir. Además cada sector contiene 512 direcciones, por tal motivo es que solo mandamos a escribir a la SD cuando se han juntado 512 valores de cada parámetro.

No se logra efectuar la telemetría como tal en tiempo real, pero por medio del almacenamiento en la tarjeta SD se logra tener la información relevante del carro para que el equipo en *pits* la analice.

Cuando el vehículo llega a *pits* a revisión la tarjeta SD es desprendida de la tarjeta EVK1100 y se inserta en una laptop. Dicha laptop funciona como un servidor de red, los otros dispositivos móviles pueden acceder a la información leída de la SD al entrar a una dirección web definida. Los datos son desplegados mediante gráficas, cada usuario puede elegir de manera independiente qué gráfica desea ver, puede seleccionar y ver más de una gráfica al mismo tiempo, ampliarla y minimizarla. En el capítulo destinado los resultados se muestran fotografías del funcionamiento y la interfaz para los dispositivos móviles.

Para la parte de *pits* se desarrolló una aplicación web principalmente con los lenguajes de programación HTML5 (*HyperText Markup Language version 5*), *Javascript*, PHP5 (*Hypertext Pre-Processor*) y XML. Como el objetivo es que la aplicación pueda visualizarse en un dispositivo móvil como el iPad, se pensó en esta solución, puesto que nos da la facilidad de desarrollar sólo una aplicación que podrá ser visualizada desde cualquier dispositivo móvil con *Wi-Fi* y un navegador.

La interfaz de usuario desarrollada se muestra a continuación:



Imagen 19. Interfaz gráfica de los dispositivos móviles de *pits*

Como se puede apreciar, la aplicación permitirá visualizar los datos almacenados en el tiempo, dentro de una tarjeta SD, durante la carrera para analizar distintos parámetros del vehículo, como son: corriente, voltaje y potencia tanto de la celda de combustible de hidrógeno, como del motor y los supercapacitores. Cabe resaltar la última opción de este menú: *Multiple Selections*, la cual permite al usuario seleccionar qué parámetros desea que se le desplieguen al mismo tiempo.

### 3.3 Diseño del programa (*software*)

Ahora se explicarán todos los programas que realizamos para el funcionamiento del sistema. Se usaron diversos programas y lenguajes de programación para las distintas aplicaciones. Se explicarán todas con diagramas de flujo generales de los programas.

#### 3.3.1 Programación del microcontrolador UC3A0512 para la adquisición de datos y el almacenamiento de información en la tarjeta SD

Para poder programar el microcontrolador UC3A dentro de la EVK1100 es necesario hacer uso de *software* de programación específico. AVR32 Studio es el programa que utilizamos, el cual tiene un ambiente de desarrollo integrado (IDE) para crear aplicaciones. Incluye un editor en lenguaje C/C++.

Lo que se hace en el programa del microcontrolador es configurar cada módulo previamente explicado y dar las instrucciones de operación y ejecución del programa principal que hace posible el correcto funcionamiento del sistema. A continuación se presentan diagramas de flujo generales del programa del microcontrolador.

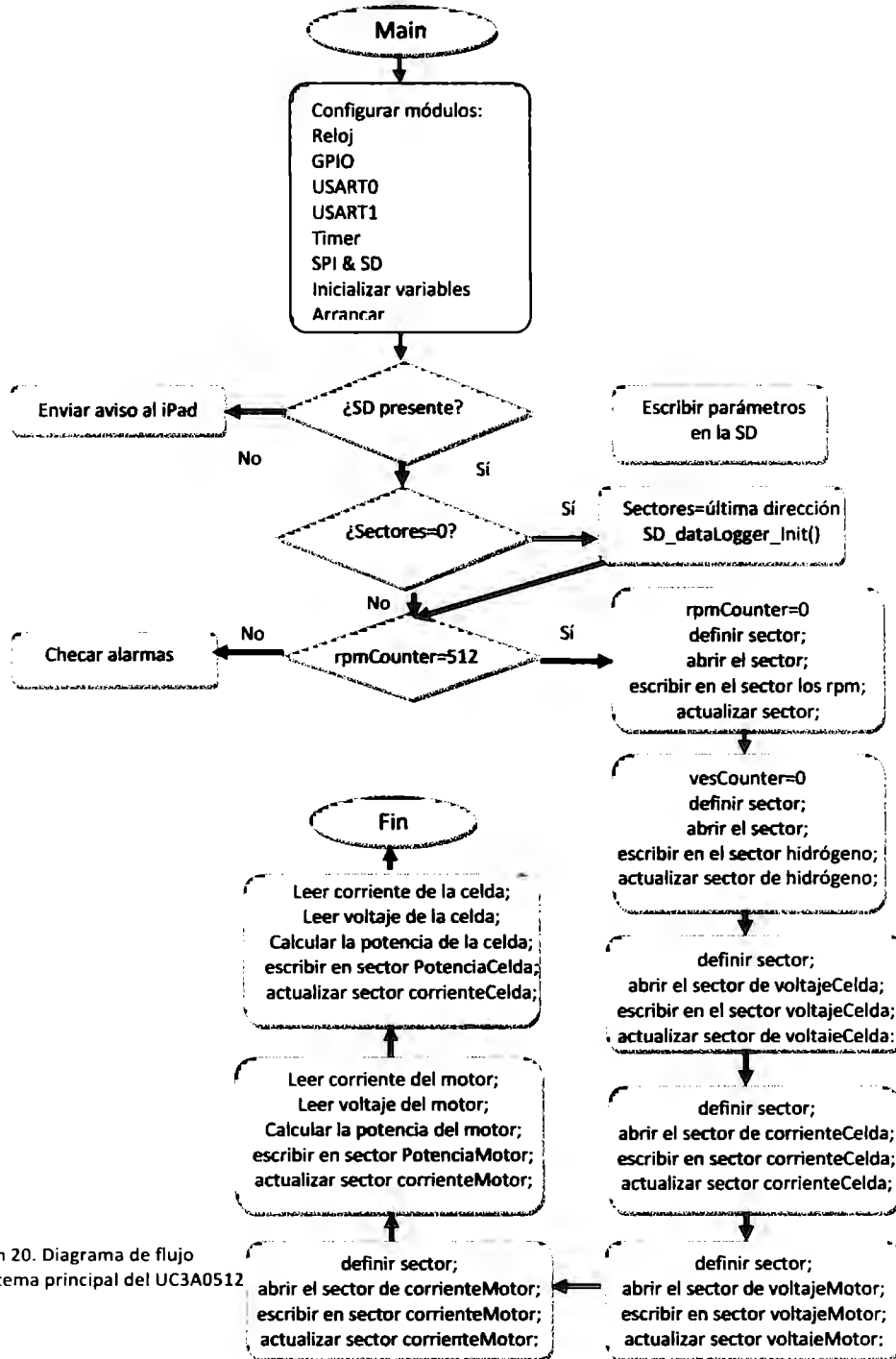


Imagen 20. Diagrama de flujo del sistema principal del UC3A0512

La función *check\_alarm()* o checar alarmas como se escribe en el diagrama de flujo, revisa si algún parámetro rebasa los valores permitidos, en caso de que se rebase, se envía una alerta al iPad.

La función *SD\_dataLogger\_Init()* se encarga de preparar los apuntadores a los segmentos de memoria en los cuáles se encontrarán almacenadas las muestras obtenidas de la etapa de control y medición de rpm. Los apuntadores corresponden a las siguientes variables:

- Rpm
- Velocidad
- Hidrógeno
- Voltaje de la celda
- Corriente de la celda
- Voltaje del motor
- Corriente del motor
- Potencia de la celda
- Potencia del motor

El puerto serial está configurado para generar una interrupción cuando recibe datos, por lo tanto cuando recibimos los datos provenientes del microcontrolador dedicado al control de la celda de combustible y del motor nosotros interceptamos la interrupción y almacenamos los datos ya sea para desplegarlos en el panel de instrumentación o para enviarlos al módulo serial/Wi-Fi.

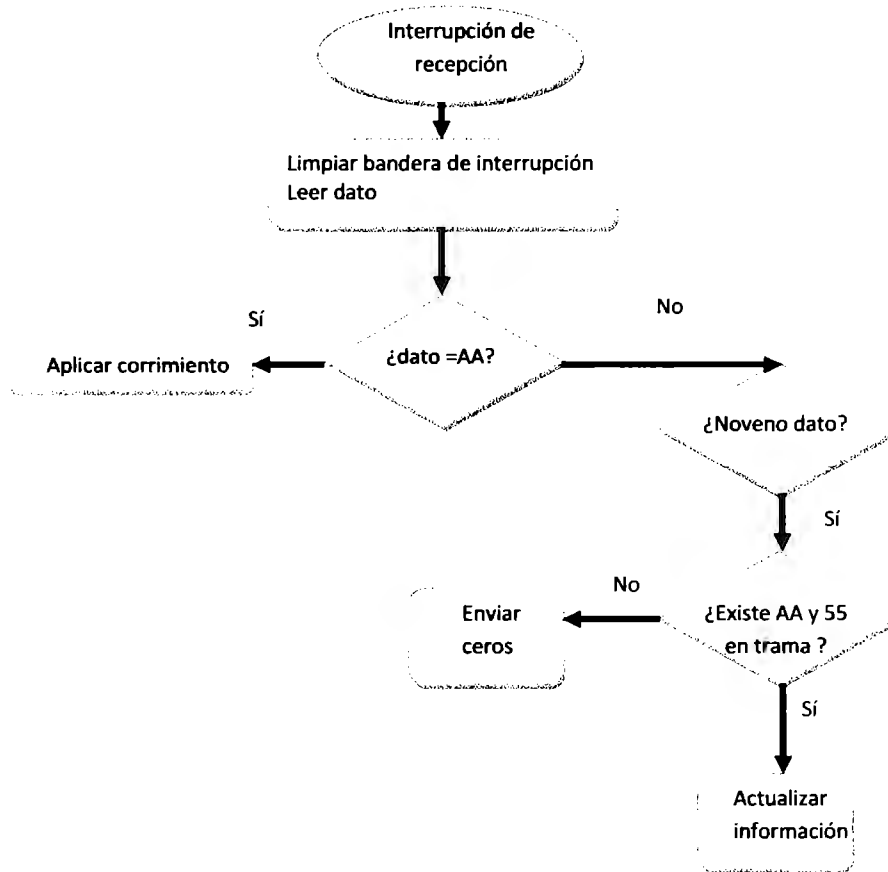


Imagen 21. Diagrama de flujo de la recepción de datos de la etapa de control

Tenemos configurado un *timer/counter* que genera una interrupción cada segundo. Cada segundo se manda por uno de los puertos seriales (conectado al módulo serial/*Wi-Fi*) del microcontrolador si hay presencia de hidrógeno y la velocidad con el objetivo de que esta información se envíe por *Wi-Fi* hacia el iPad.

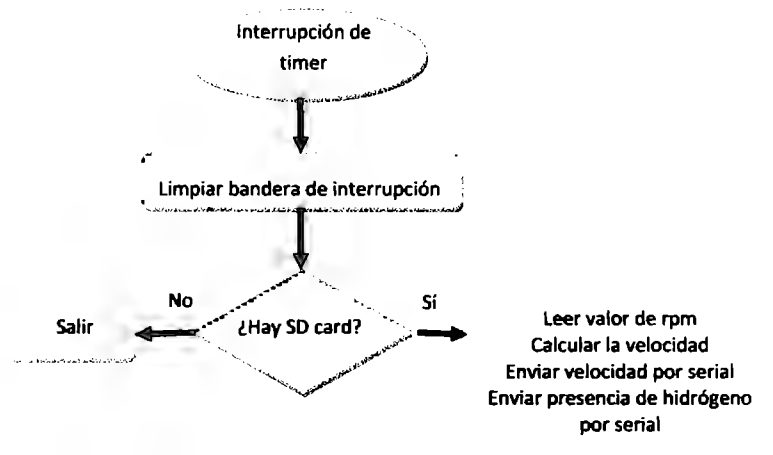


Imagen 22. Envío de datos al módulo RS232/*Wi-Fi*

Estos son los diagramas de flujo muy generales y principales del programa. No hay que olvidar que se debe hacer la configuración de cada módulo e inicializarlo.

Módulo del microcontrolador	Configuraciones
<b>USART 0 (Serial/<i>Wi-Fi</i>)</b>	<b>Baudrate: 38 400</b> Tamaño de palabra: 8 No bit de paridad Un bit de stop
<b>USART 1 (Entrada de control)</b>	<b>Baudrate: 2400</b> Tamaño de palabra: 8 No bit de paridad Un bit de stop
<b>GPIO</b>	Habilitar pin RX para el USART0( Rabbit (Serial/ <i>Wi-Fi</i> )) Habilitar pin TX para el USART0 (Rabbit (Serial/ <i>Wi-Fi</i> )) Habilitar pin RX para el USART1 (microcontrolador) Habilitar pin Tx para el USART1 (microcontrolador) Habilitar pines del SPI: reloj, MOSI, MISO y NPCS

Tabla 9. Configuraciones de los módulos del UC3A0512

El fragmento de código que hace posible la configuración del `usart0` y del `usart1` está definido en un método llamado `usart_setup()`

### 3.3.2 Programación del módulo RS232/Wi-Fi

En este código lo que se hace es primeramente la configuración del módulo para que pueda recibir datos por serial a determinado *baudrate* y transmitir dichos datos por *Wi-Fi* es necesario definir ciertos parámetros como las direcciones IP, máscaras e identificadores de servidores.

A continuación se describen algunas de las funciones principales para la configuración y ejecución.

#### Método *ifconfig()*

Este método manda y captura los parámetros de la red en tiempo de corrida. Se recibe o manda un número arbitrario de parámetros. Configura la dirección IP, la máscara, el enrutador y los nombres de los dos servidores involucrados, finalmente la interface se activa.

Estructura del método:

*Ifconfig*(interface, máscara, enrutador, nombre del servidor1, nombre del servidor 2, activar, terminar lista)

Parámetro	Descripción
<b>IF_WIFI0</b>	Interfaz <i>Wi-Fi</i> 0
<b>IFS_IPADDR</b>	Establece la dirección IP local
<b>IFS_NETMASK</b>	Establece la máscara de red
<b>IFS_ROUTER_SET</b>	Establece el enrutador por defecto
<b>IFS_WIFI_SSID</b>	Establece el SSID para el dispositivo <i>Wi-Fi</i>
<b>IFS_WIFI_ROAM_ENABLE</b>	Establece el <i>roaming</i> como encendido o apagado
<b>IFS_WIFI_ROAM_BEACON_MISS</b>	Número de paquetes que desea desechar antes de iniciar el punto de acceso.
<b>IFS_WIFI_MODE</b>	Elige el modo de red, si es <i>Ad-hoc</i> o infraestructura
<b>IFS_END</b>	Da fin a la lista de parámetros

Tabla 10. Descripción de parámetros de configuración para *Wi-Fi*

Los parámetros tienen los siguientes valores

Dirección IP	10.10.6.100
Máscara de red	255.255.255.0
Dirección del enrutador	10.10.6.1
Nombre de servidor 1	10.10.6.1
Nombre del modem	Tonatiuh MyWi
Modo	Ad-hoc

El SSID (*Service Set Identifier*) es un nombre que permite identificar a los paquetes en una red inalámbrica *Wi-Fi* como parte de esa red. Este identificador consta de 32 caracteres generalmente alfanuméricos. Si un equipo inalámbrico desea comunicarse con otro dentro de la red, estos dispositivos deben compartir un mismo SSID. En una red *ad-hoc* se utiliza la BSSID (*Basic SSID*).

**Método *ifpending*(Interfaz)**

Este método lo que hace es regresar una indicación acerca del estado de la interfaz seleccionada, en este caso de *Wi-Fi*. Su respuesta da más información que el estatus, el cual solo especifica si está activa o no la interfaz. Los valores que puede regresar son los siguientes.

Valor	Descripción
0	Interfaz inactiva y no pendiente
1	Interfaz inactiva y pendiente
2	Interfaz activa y no pendiente
3	Interfaz activa y pendiente

Como parámetro se envía el identificador de la interfaz cuyo estatus se desea conocer, en nuestro caso enviamos el identificador del *Wi-Fi*.

**Método *tcp\_listen*()**

Esta función le dice al DCRTCP que se aceptará una sesión por un puerto en específico. Los parámetros que este método recibe son los siguientes:

- Socket
- Puerto a atender
- La dirección IP a la que se debe conectar
- Puerto con el cuál establecer la conexión
- Nombre de la función a llamar cuando se reciba un dato
- Palabra reservada

Este método regresa un "1" si fue exitoso y un "0" si hubo un error.



### Programa principal

El siguiente diagrama de flujo muestra el proceso que se realiza para poder establecer una comunicación *Wi-Fi* entre el *rabbit* y el iPad. El programa para el microcontrolador *rabbit* 5000 está hecho en lenguaje C y programado en *Dynamic C*.

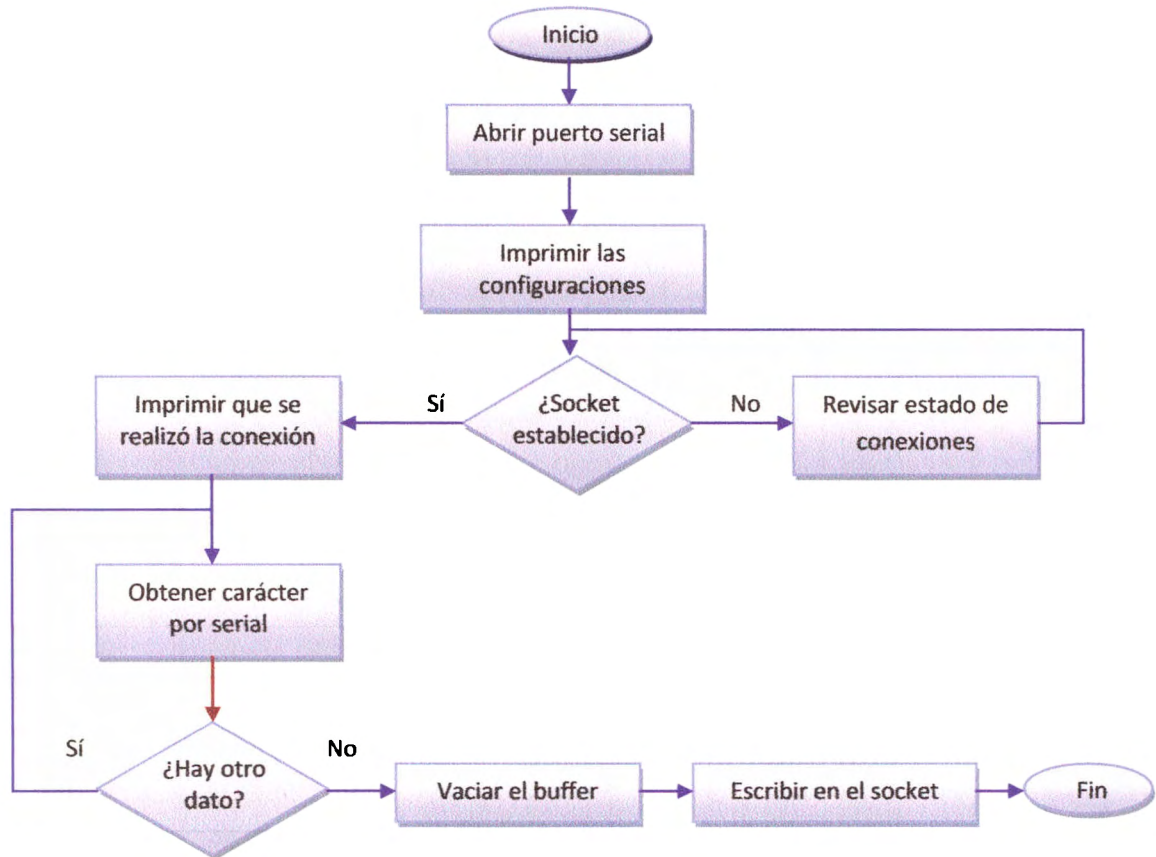


Imagen 23. Diagrama de flujo del programa principal del rabbit 5000

### 3.3.3 Programación del sistema de archivos para lectura de tarjeta SD

Debido a que la tarjeta de adquisición de datos guarda los la información recibida, calculada y medida sin obedecer ningún sistema de archivos, fue necesario realizar una aplicación que lograra recuperar los datos contenidos en la tarjeta.

Por otro lado, la aplicación web que se realizó, necesita recibir los datos en un formato especial y compatible con las tecnologías utilizadas. Por este motivo, la aplicación de lectura de la tarjeta SD resultó una capa de compatibilización importante entre la tarjeta de adquisición de datos y la aplicación web.

El programa fue desarrollado en el lenguaje C# y utiliza ampliamente librerías del .NET Framework y del núcleo de Windows, ubicadas en el archivo *Kernel32.dll*.

Las librerías del núcleo, permiten efectuar las siguientes tareas:

- Obtención de información sobre discos conectados al sistema
- Obtener acceso de lectura a un disco físico
- Realizar lecturas a nivel de bit de algún disco físico conectado al sistema

Las librerías más significativas utilizadas del *.NET Framework* permiten:

- Tener una aplicación con entorno amigable
- Crear documentos HTML
- Manejo de archivos a nivel de sistema operativo

Las funciones más importantes implementadas por el programa son:

- Recuperación de datos en memoria SD sin sistema de archivos
- Generación de documentos XML, que cumplen con la especificación de la librería estandarizada en JavaScript JChart para la generación y el trazado de gráficas.
- Selección de carpeta donde se desea guardar los documentos XML generados
- Selección de unidad de disco desde la cual se leerán los datos guardados por la tarjeta SD

A continuación se describe la interfaz del programa

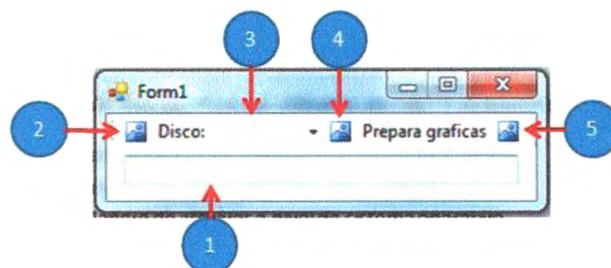


Imagen 24. Interfaz gráfica de la aplicación para lectura de tarjeta SD

1. Cuadro de texto dedicado a la ruta donde se guardarán los documentos XML generados por el programa
2. Este botón permite obtener una ventana que nos permite navegar a través del sistema de archivos y seleccionar la carpeta donde se guardarán los documentos XML
3. Cuadro de selección del disco origen.
4. Generación de documentos XML
5. Botón de propósitos de monitoreo, este botón muestra el sector 13 del disco seleccionado para observar un patrón específico.

### 3.3.4 Programación de la interfaz gráfica para los dispositivos móviles

Para poder realizar la aplicación web para *pits* se emplearon diversos lenguajes de programación HTML5, *Javascript*, PHP5 y XML.

HTML5 consta de estándares para el diseño y desarrollo de páginas web. Permite definir la manera en la que se presenta la información en el explorador de internet y cómo el usuario puede interactuar con ella. Además HTML5 facilita mayor contenido media (video y audio principalmente).

*Javascript* es un lenguaje de programación que permite a los desarrolladores crear acciones en las páginas web. Es un lenguaje utilizado para crear programitas que posteriormente pueden ser insertados en una página web, de esta forma se propician diversas maneras de interactuar con las páginas.

PHP5 es otro lenguaje de programación diseñado en principio para la creación de páginas web. En la actualidad se usa sobre todo para la interpretación por parte del servidor. La ventaja de este lenguaje es que puede desplegarse en la mayoría de los servidores web y en casi cualquier sistema operativo y plataformas sin costo.

XML o lenguaje extensible de marcas es un conjunto de reglas que permite al usuario tener su información estructurada jerárquicamente por medio de etiquetas o *tags* creadas por el mismo usuario.

La librería *Jquery Mobile*, la cual está programada principalmente en *Javascript* y tiene algunas funciones implementadas por HTML5, permite conseguir una vista que se ajusta automáticamente a la resolución del dispositivo móvil, esto nos es de importancia ya que se planea tener diversos dispositivos móviles en *pits* y no únicamente un iPad.

A continuación se presenta el diagrama de flujo muy general de lo que consiste el programa para la aplicación web de *pits*.

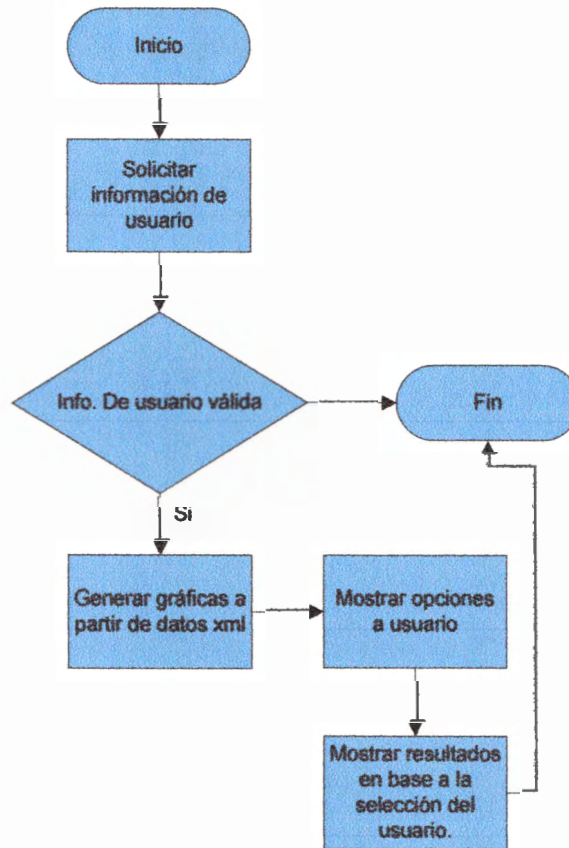


Imagen 25. Diagrama de flujo de la aplicación web para *pits*

En el programa, luego de definir las variables, el primer paso es la validación del nombre de usuario y de la contraseña; el usuario debe ingresar estos valores. Si los valores ingresados son los correctos entonces se conecta a la siguiente locación "Location: <http://10.48.175.102/cleanmobilities/pits.html>"<sup>1</sup>, la cual despliega un menú. Sin embargo, si los valores ingresados por el usuario no son los correctos, se manda a otra página que indica que hay un error con la validación y no se puede visualizar el menú.

<sup>1</sup> El equipo encargado del servicio y mantenimiento del vehículo situado en *pits* se debe conectar a la dirección web <http://10.48.175.102/cleanmobilities/pits.html>

La siguiente parte del código no se muestra en el diagrama de flujo pero se basa en la configuración de la interfaz gráfica, definición de colores, el tipo de letra, ubicación de los parámetros en la pantalla, etc.

Cuando el sistema realiza la correcta validación de la información dada por el usuario, en cada dispositivo móvil se hace el procesamiento para poder generar las gráficas correspondientes a todos los valores, dependiendo de la lectura que se hace de la tarjeta SD desde la laptop. Posteriormente el programa espera las entradas del usuario, en éstas se seleccionan las gráficas que se desean visualizar, para finalmente desplegarlas en pantalla.

### 3.4 Alternativas de Solución

Antes de tomar las decisiones finales para la implementación del sistema final con las tecnologías y dispositivos seleccionados ya se habían pensado en otras opciones que podrían dar solución a la problemática presentada.

Para la comunicación inalámbrica entre la tarjeta de adquisición de datos y el iPad se pensó en una comunicación vía *bluetooth* como primer opción. El protocolo de *bluetooth* parecía ser el más óptimo por su seguridad, bajo costo y por el área de cobertura (no se necesitaba mucho alcance), sin embargo aún no se puede establecer una comunicación bajo este protocolo entre un dispositivo de Apple y otro que no lo sea, es decir, sólo se puede hacer entre dos dispositivos Apple. Por el motivo anterior, *Wi-Fi* fue mejor y dio solución a nuestro problema.

El panel de instrumentación o tablero del vehículo no necesariamente debía ser un iPad, por ejemplo, el equipo Electratón, representante del Tecnológico de Monterrey utiliza una laptop, el conductor del carro lleva una laptop, con la tarjeta SD integrada para guardar las mediciones. Una laptop hubiera podido desempeñar bien el papel de un tablero, sin embargo se descartó la opción debido a que sería incómodo para el conductor y maximizaría el peso.

También se pudo haber utilizado una matriz de *leds* o una pantalla LCD que desplegara los datos de manera digital. La funcionalidad se cumple, no requiere de mucho espacio y no es pesado, es una buena opción. La razón por la que no se escogió ésta opción fue por la innovación que presenta tener un iPad como tablero dentro de un vehículo, aprovechamos el fuerte impulso que existe actualmente hacia el desarrollo de aplicaciones para dispositivos móviles, sobre todo para dispositivos de Apple.

## Capítulo 4. Pruebas y resultados

En este capítulo se explican las distintas pruebas que se realizaron antes de llegar al sistema final. Las pruebas involucran a los medios de comunicación inalámbrica entre la tarjeta de adquisición de datos y el iPad, cómo se logró establecer una conexión y cómo hacer la transmisión y recepción de datos por dicho medio inalámbrico. También se hicieron pruebas del circuito encargado de la medición de rpm's y del servidor web. En este capítulo se incluyen diagramas de las conexiones y fotografías de los resultados.

### 4.1 Recepción de datos por serial

Una de las pruebas que se realizaron fue la recepción de datos por parte del equipo destinado a desarrollar la etapa de control de la celda de hidrógeno y del motor del vehículo.

Para poder sincronizar los dos proyectos y microcontroladores fue necesario definir una trama de envío de datos, la trama consta de un "AA" al principio y un "55" al final, en el interior de la trama van contenidos los valores de la etapa de control. Este método simple permite reconocer si una trama se envió adecuadamente o presenta errores.

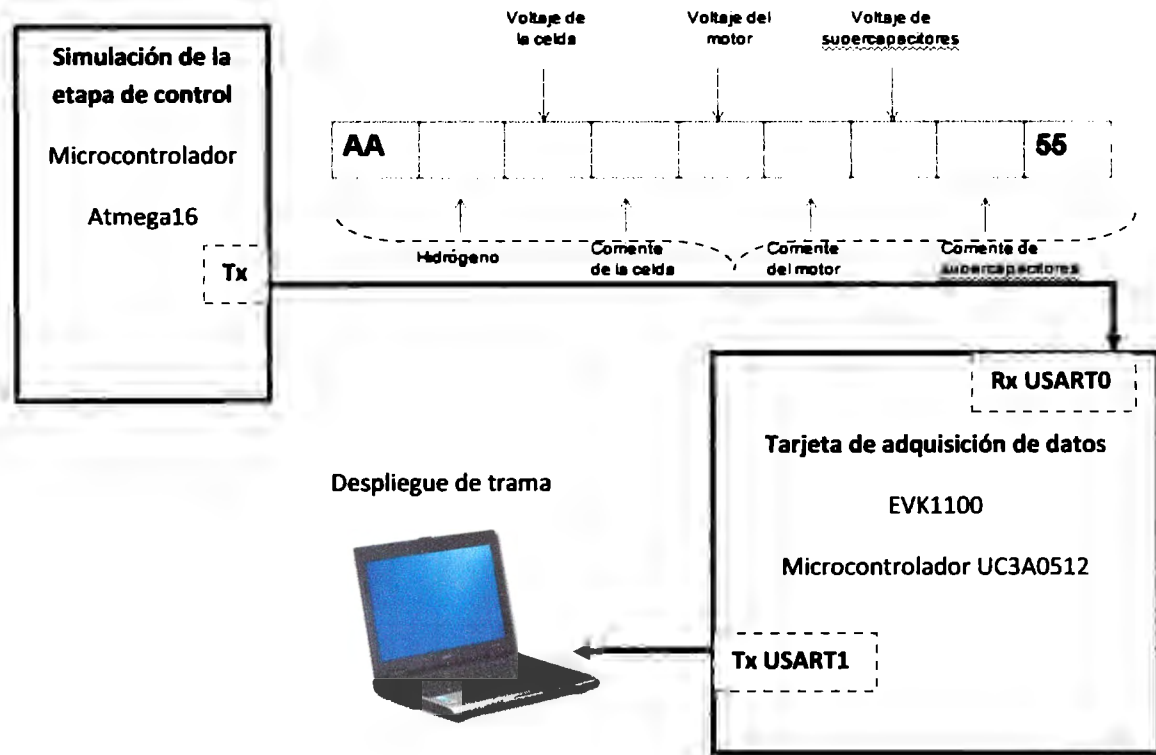
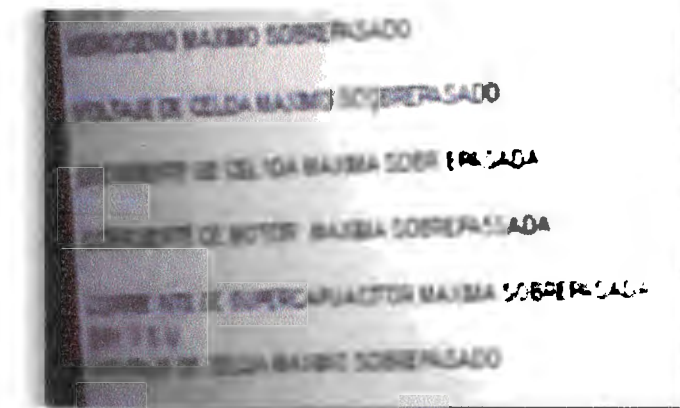


Imagen 26. Diagrama del envío de trama entre etapa de control y tarjeta de adquisición de datos

Para poder hacer las pruebas de la recepción de datos se programó en un microcontrolador una trama que simula el envío de datos de la etapa de control. La manera de revisar esta comunicación fue por medio de la visualización de dichos datos en la computadora. Se conecta directamente un cable que va del pin de transmisión serial del Atmega 16 al pin de recepción de uno de los dos puertos seriales del UC3A0512. Luego de revisar si los datos recibidos fueron correctos, se mandaba hacia la computadora la información por medio de otro puerto serial de la tarjeta EVK, ya que la computadora recibe los datos, estos son desplegados en pantalla usando *hyperterminal*.

Si la trama se recibe correctamente, se almacenan los datos, si no ocurre así se guarda el valor anterior.

La siguiente imagen es una fotografía tomada directamente a la pantalla de la computadora. La computadora está corriendo *hyperterminal* con un *baudrate* de 2400bits/seg, valor con el que el microcontrolador de la etapa de control transmite sus datos.



Los valores programados en el microcontrolador son constantes. Para observar el correcto funcionamiento del sistema de alarmas se programaron en el microcontrolador algunos valores que estuvieran fuera del rango esperado, por tal motivo en la pantalla del *hyperterminal* se observa que algunas variables están "sobrepasadas".

Imagen 27. Prueba de la recepción por serial en la EVK1100

El código programado en el ATmega16 (simulador de la etapa de control) se basa únicamente en el envío de datos por serial y en la configuración de los puertos y pines a usar.

El puerto serial está configurado con 8 bits de datos, un bit de stop y no paridad, trabaja con un *baudrate* de 2400 bits/seg. El programa principal manda constantemente por serial por medio de la función *putchar()*.

```

// USART initialization
// Communication Parameters:
8 Data, 1 Stop, No Parity
// USART Receiver: Off
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 2400
UCSRA=0x00;
UCSRB=0x08;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0xCF;

while (1)
{
    delay_ms(20);
    putchar(0xAA);
    putchar(0x30);
    putchar(0x31);
    putchar(0xFF);
    putchar(0x33);
    putchar(0xFF);
    putchar(0x35);
    putchar(0xFF);
    putchar(0x55);
}
    
```

Imagen 28 Código de configuración del puerto serial

## 4.2 Pruebas del muestreador

Para probar esta etapa del proyecto conectamos el sensor óptico a la rueda, la salida del sensor se conecta directamente al circuito *Schmitt Trigger*, el cual genera los pulsos cuadrados a la salida dependiendo la entrada que reciba del sensor. La salida del *Schmitt Trigger* es medida con una punta del osciloscopio y es enviada a un pin de recepción del USART del microcontrolador UC3A512. Además para corroborar los datos recibidos, estos mismos son enviados hacia otro puerto serial del microcontrolador, éste se conecta mediante un cable Serial/USB hacia la laptop que despliega por *hyperterminal* los datos recibidos.

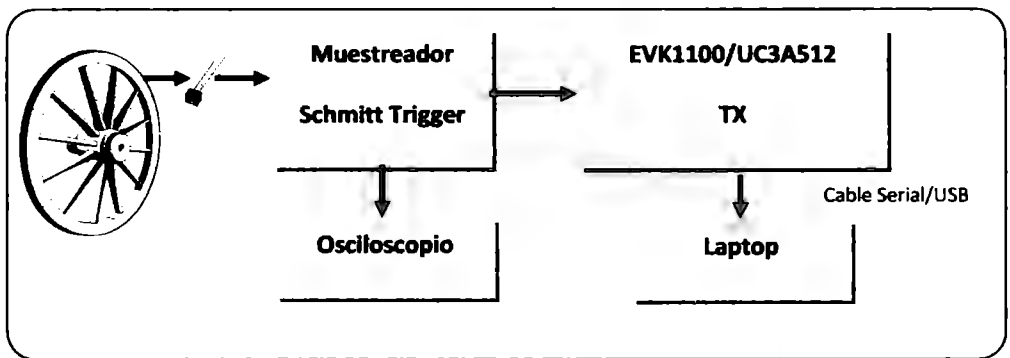


Imagen 29. Diagrama a bloques de la conexión para la prueba del muestreador



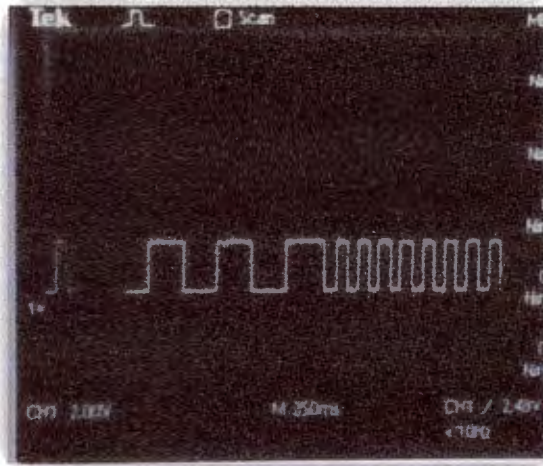


Imagen 30. a) Respuesta del digitalizador

Si la rueda gira más rápido el ancho disminuye y la frecuencia aumenta y viceversa, si la rueda disminuye su velocidad el ancho de pulso aumenta y la frecuencia disminuye.

La imagen muestra una fotografía tomada directamente del osciloscopio, se visualiza el pulso ya digitalizado, se puede observar que unos pulsos presentan un ancho mayor que otros, esto es debido al incremento de la velocidad de la rueda.

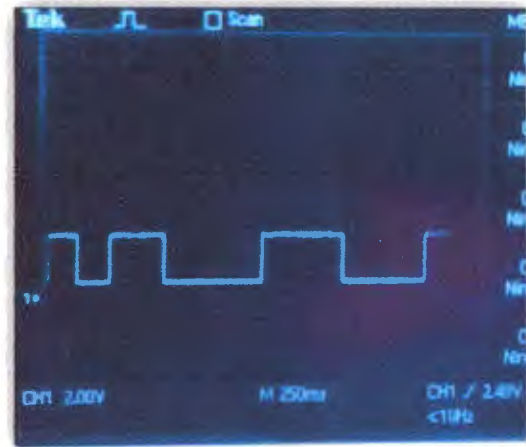


Imagen 31. b) Respuesta del digitalizador

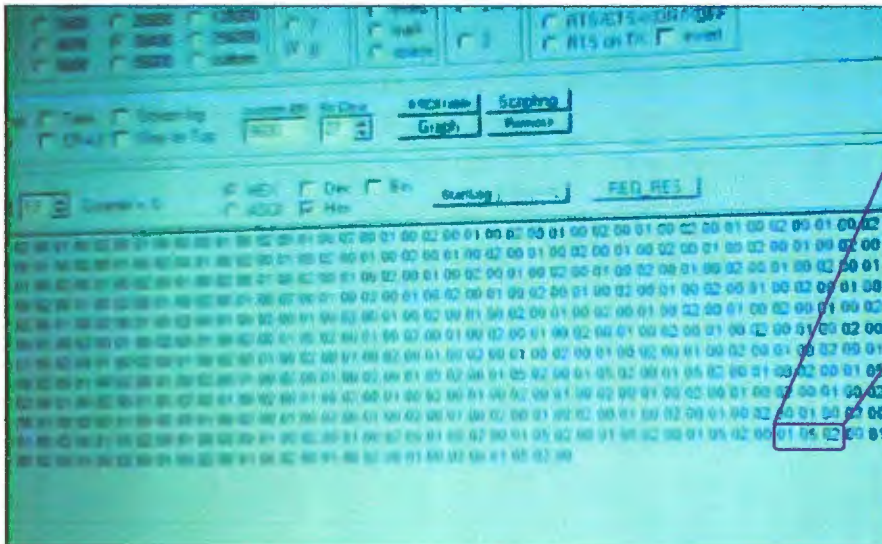


Imagen 32. Recepción por serial de la velocidad de la llanta

En la imagen se puede observar que la velocidad se incrementa respecto a valores anteriores.

### 4.3 Escritura de datos en la tarjeta SD

Cuando ya se tuvo la adquisición de datos (aunque fuera simulada) se pudo probar la escritura de datos en la tarjeta SD. Para realizar este paso se repitió la prueba anterior para conocer qué valores se estaban recibiendo y posteriormente corroborar que fueran los que se escribieran en la tarjeta.

Cada 15ms se hace la recepción de datos por parte del microcontrolador de la etapa de control, cada dato recibido se va guardando en un arreglo.

Es importante señalar que las tarjetas SD se componen de sectores, cada sector es capaz de guardar 512 datos en código hexadecimal. Además cada que se quiere hacer una escritura en la tarjeta se escribe el sector completo, por tal motivo lo que hicimos fue esperar a juntar 512 valores correspondientes a cada parámetro antes de realizar la escritura en la SD.

La tarjeta se dividió de una manera específica para poder almacenar organizadamente los datos, por ejemplo los primeros 2000 sectores corresponden a la escritura del voltaje de la celda, los siguientes 2000 sectores corresponden a la corriente de la celda y así sucesivamente.

Para corroborar que efectivamente los datos recibidos fueran los mismos que los que se escribían en la tarjeta, nos apoyamos de un programa llamado HxD el cual permite hacer este tipo de procesos. El programa ofrece una interfaz sencilla de utilizar y facilita la visualización del contenido de cada sector.

La siguiente imagen representa la interfaz gráfica del programa HxD.

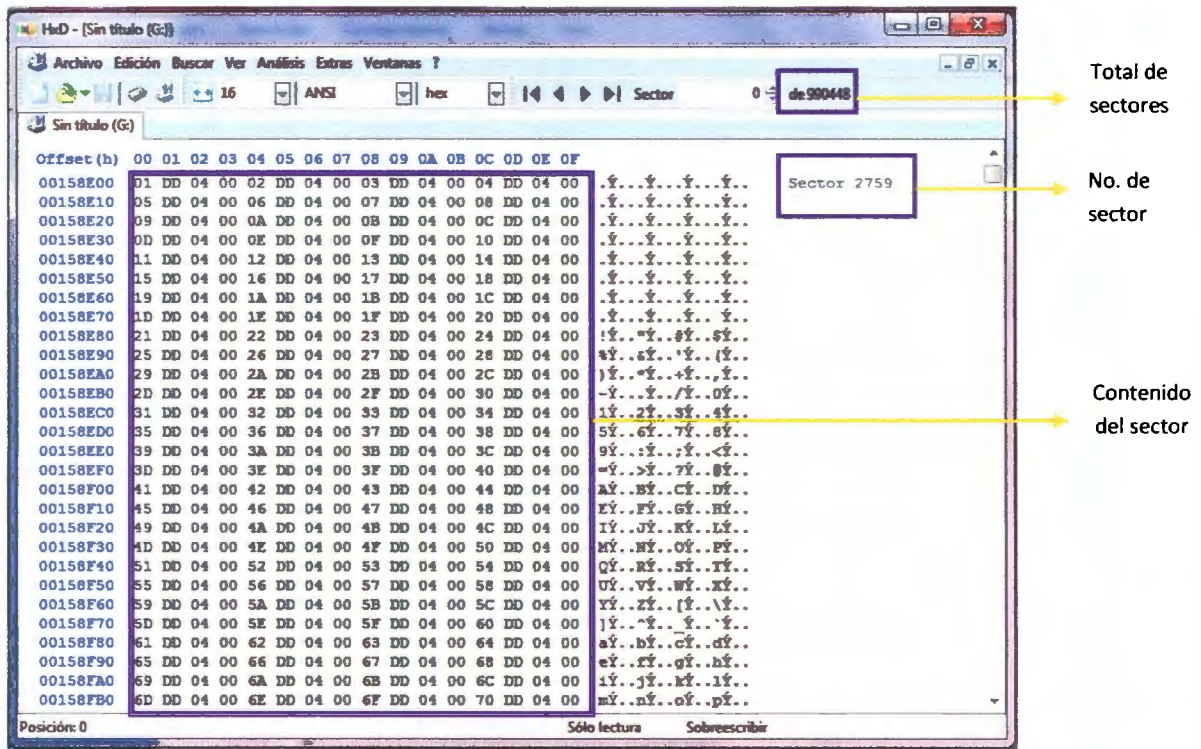


Imagen 33. Lectura de un sector de la tarjeta SD desde el programa HxD

En la imagen 33 se observa lo que se recibe por serial, estos datos son enviados por *Wi-Fi* al iPad, sin embargo para probar el funcionamiento, se manda a la *hyperterminal* de la computadora toda esta información por serial. En la pantalla se visualiza que algunos de los parámetros envían alarmas y en otros solo el valor.

#### 4.4 Comunicación inalámbrica entre la tarjeta de adquisición de datos y el iPad

Sin duda una de las etapas más difíciles del proyecto fue establecer la conexión y el intercambio de información entre la tarjeta EVK1100 y el iPad (para el despliegue de datos dentro del vehículo). Es importante mencionar en este apartado que la primera opción que se tuvo para establecer esta comunicación fue por medio del *bluetooth*.

Bluetooth parecía ser la solución más apropiada al problema debido a su infraestructura simple para el funcionamiento, bajo costo del equipo, área de cobertura que abarcaba (no se necesitaba una cobertura muy amplia pues la comunicación era exclusivamente para el interior del carro) y seguridad que garantiza en el envío y recepción de información. Por todos estos factores mencionados, lo primero que se realizó fue una comunicación por *bluetooth* por medio de un módulo GL-6B el cual recibe información por RS232 y la radia por *bluetooth*.

Los avances que se lograron fueron el establecimiento de la conexión, reconocimiento del dispositivo bluetooth, recepción de datos por serial y envío por *bluetooth* hacia una computadora.



La imagen muestra que una computadora identificó al módulo serial/*bluetooth* con el nombre *Clean Mobilities* y pudo agregarlo como dispositivo Bluetooth.

Imagen 34. Identificación de dispositivo bluetooth

Sin embargo, a pesar de los grandes avances que se lograron, fue imposible lograr el intercambio de información entre la tarjeta EVK1100, módulo y iPad. Un iPad sólo puede comunicarse por *bluetooth* con otro dispositivo de Apple. Se recurrió a diversas fuentes y propuestas, pero todos fueron intentos fallidos, es por eso que se decidió cambiar la tecnología de comunicación inalámbrica a *Wi-Fi* la cual aseguraba que se estableciera la conexión e intercambio de datos con el iPad.

## 4.5 Prueba de la lectura de la SD

Las últimas pruebas que se hicieron del funcionamiento del sistema consistieron en la lectura de la tarjeta SD dentro de una computadora o laptop, de esta forma se prueba que la información almacenada en la tarjeta en un formato crudo, es decir con puros unos y ceros almacenados sin ser interpretados aún, puede ser leída y transformada a un formato estándar (xml).



Imagen 35. Inserción de la tarjeta SD en la laptop

A continuación se presenta a manera de manual de usuario los pasos que se siguen para poder leer datos contenidos en la SD una vez que ésta se introduce a la laptop en *pits*.

1. Abra la aplicación `CleanMobilitiesPitsSDReader.exe`, para encontrarla, diríjase a la ruta `CleanMobilitiesPitsSDReader/bin/Debug`, dentro de la carpeta del proyecto

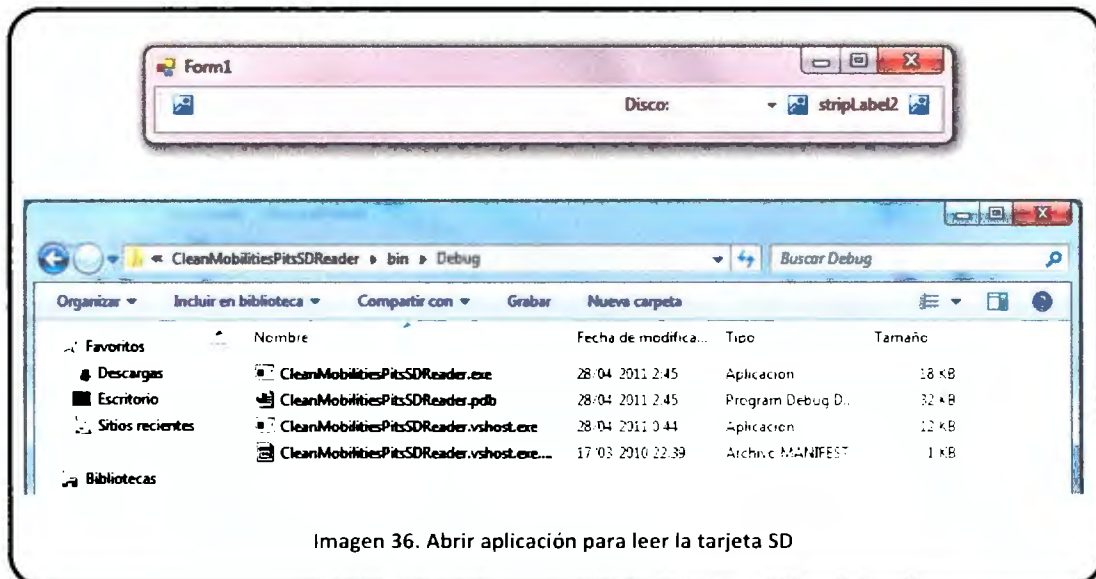


Imagen 36. Abrir aplicación para leer la tarjeta SD

2. Ya que la ventana del programa ha sido seleccionada, es necesario indicar la ruta del sistema de archivos donde se desean guardar los documentos XML generados por el programa.

La selección de esta carpeta tiene suma importancia, debido a que la misma ruta deberá estar configurada en la aplicación web para tener acceso a los datos reales de las graficas

Existen 2 maneras de seleccionar la ruta:

## Sistema de Adquisición de Datos y Telemetría para un Kart

- a. Escribir la ruta completa hacia una carpeta existente en el cuadro de texto destinado para ello.
- b. Presionar el botón Seleccionar Carpeta, donde se presenta una ventana para realizar la tarea.

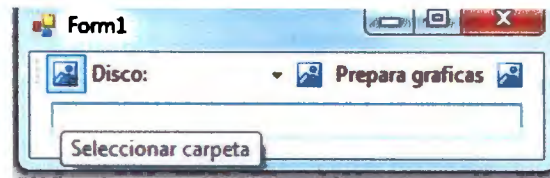


Imagen 37. Interfaz gráfica de la aplicación para leer tarjeta SD

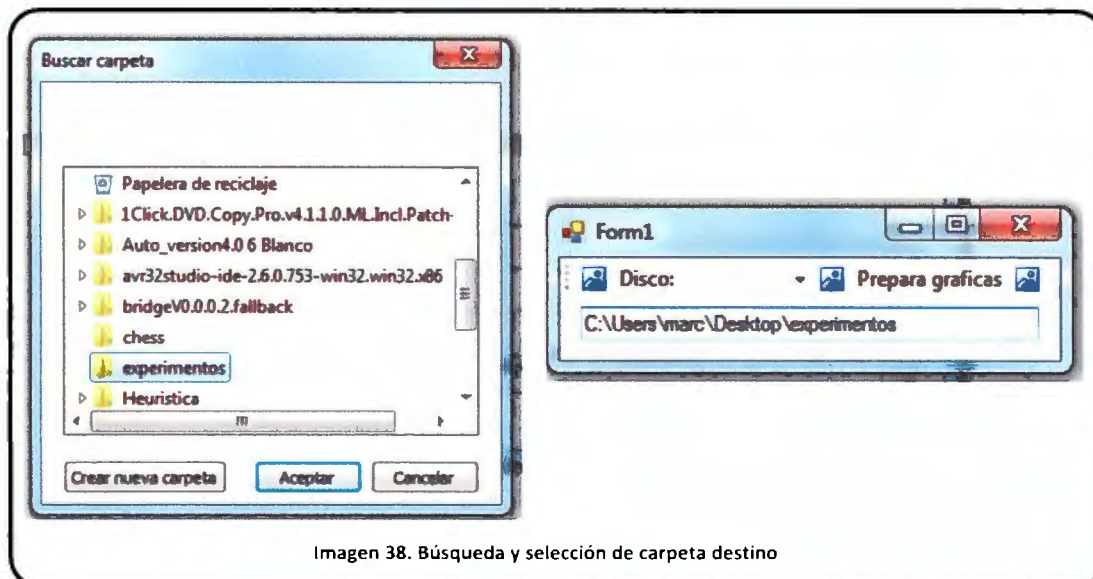


Imagen 38. Búsqueda y selección de carpeta destino

3. Después de haber seleccionado la carpeta destino para los documentos XML, podemos seleccionar la unidad de disco de la cual deseamos leer los datos generados.

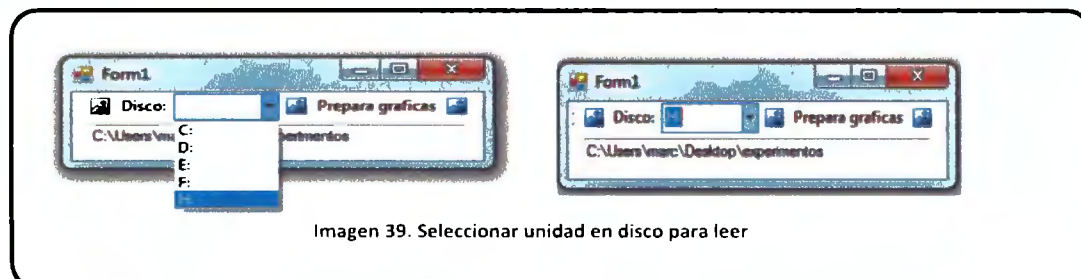


Imagen 39. Seleccionar unidad en disco para leer

4. Presionar el botón Recopilar información para recuperar los datos y traducirlos al formato XML de forma automática

5. Finalmente y como paso adicional es posible verificar la existencia de los documentos XML que el programa acaba de generar, para hacer esto, es necesario dirigirse a la ruta que se seleccionó anteriormente

## 4.6 Pruebas de despliegue de gráficas en pits

Después de programar la aplicación para que la computadora trabaje como un servidor web para dispositivos móviles terminales como iPad, iPod, Blackberry, etc. Se decidió realizar las pruebas pertinentes, primeramente con datos ficticios, únicamente para corroborar el despliegue adecuado de las gráficas y la interfaz gráfica.

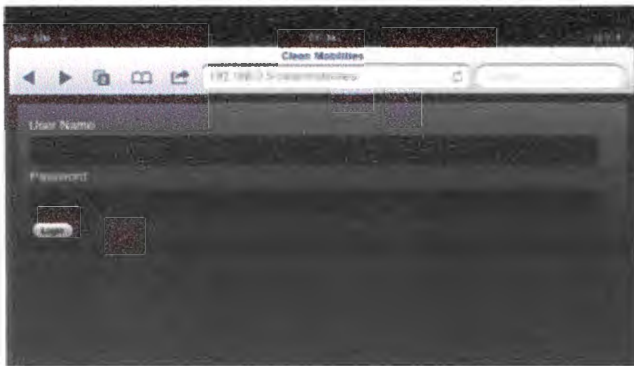


Imagen 40. Pantalla principal y registro del usuario

### Pantalla principal y registro

Ésta es la pantalla principal que se presenta para los dispositivos de *pits*. Se pide una cuenta y una contraseña, esto por cuestiones de seguridad y que sólo la gente del equipo pueda acceder a la información contenida en la tarjeta SD y proporcionada por el servidor web (laptop).

### Menú de parámetros

Una vez que se corroboraron el nombre de usuario y contraseña, se despliega un menú con los títulos de los parámetros a los que se puede acceder para observar su respectiva gráfica.

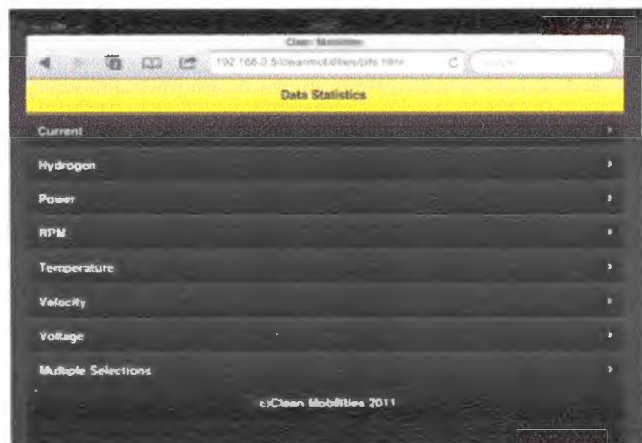


Imagen 41. Despliegue del menú

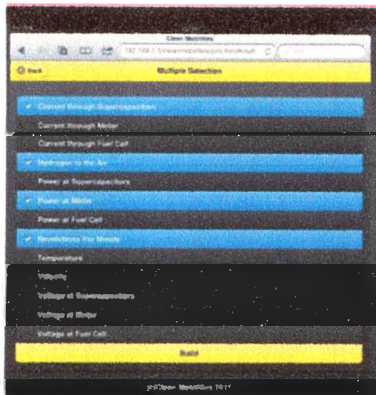


Imagen 42. Selección de parámetros a graficar

### Selección de parámetros

Se puede seleccionar más de un parámetro para visualizar las gráficas de manera simultánea

### Visualización de gráficas

Finalmente las gráficas se pueden apreciar como en la imagen de la izquierda, se observa que se puede analizar más de una gráfica a la vez.

Cada gráfica contiene en la parte inferior un deslizador, el cual permite al usuario ajustar el tamaño de cada una de las gráficas de manera individual, para así poder visualizarlas mejor según conveniencia y según el dispositivo móvil utilizado para su visualización.

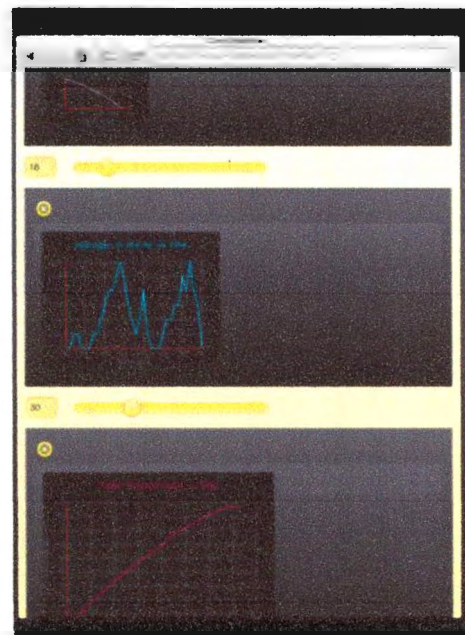
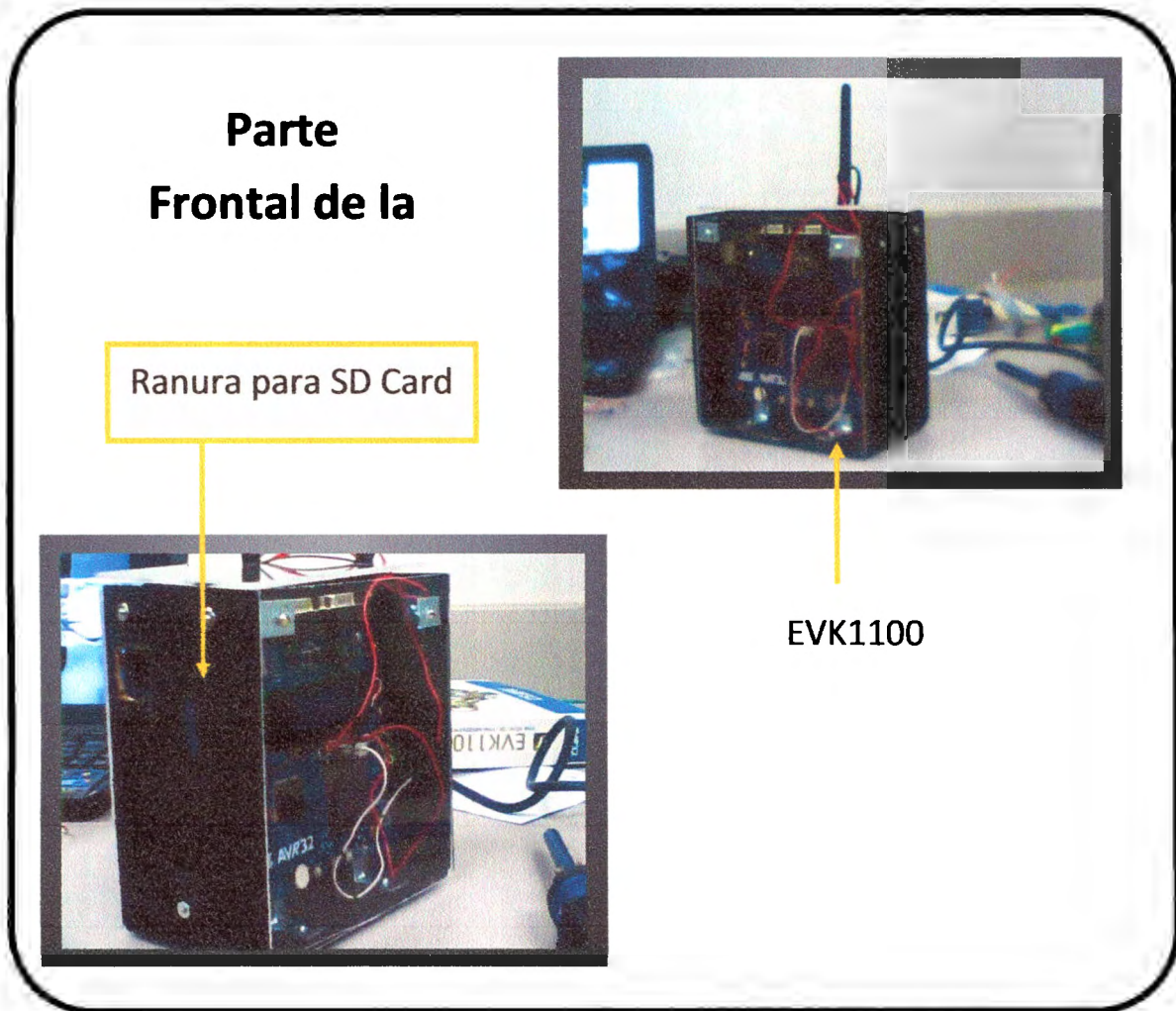


Imagen 43. Despliegue de gráficas



Por reglamento de la competencia, todos los circuitos electrónicos que estén dentro del vehículo deberán estar contenidos en una caja transparente para que se pueda visualizar. Por tal motivo construimos una caja basada en madera y acrílico, que le dé seguridad a nuestras tarjetas e instrumentación y que a la vez sea visible y práctica.

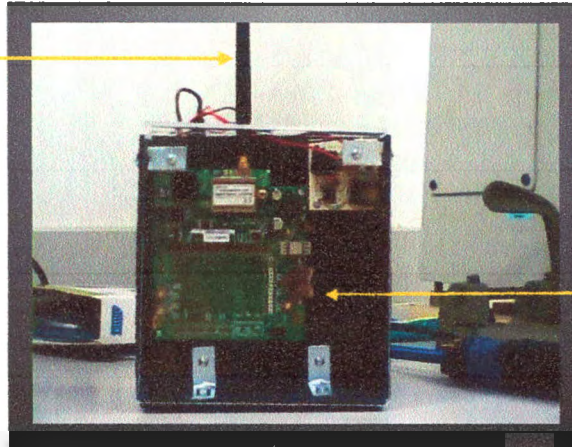
A continuación se presentan unas fotografías de la caja con la circuitería dentro de ella.



La caja se divide en tres secciones, en una se ubica la tarjeta EVK1100, en la sección de en medio se encuentran las baterías que alimentan a la EVK1100 y al *rabbit*, en la tercera sección está empotrado el circuito medidor de rpm's y la tarjeta de desarrollo del *rabbit*. Cada sección está separada por tablas delgadas de madera, en las paredes de la caja está el acrílico, cada parte con las ranuras correspondientes para la fácil manipulación de los circuitos.

**Parte Posterior  
(Rabbit y circuito medidor de rpm)**

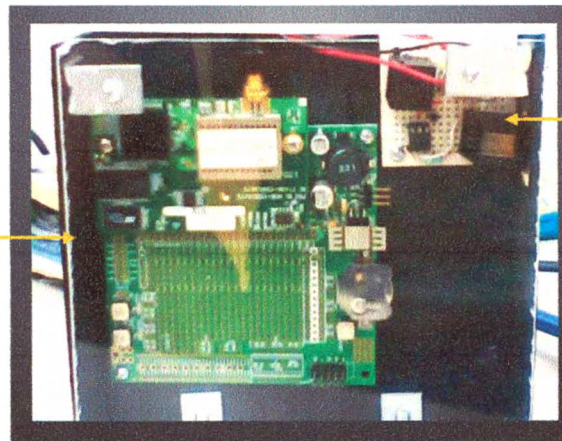
Antena para  
Wi-fi



Rabbit  
(Serial/Wi-fi)

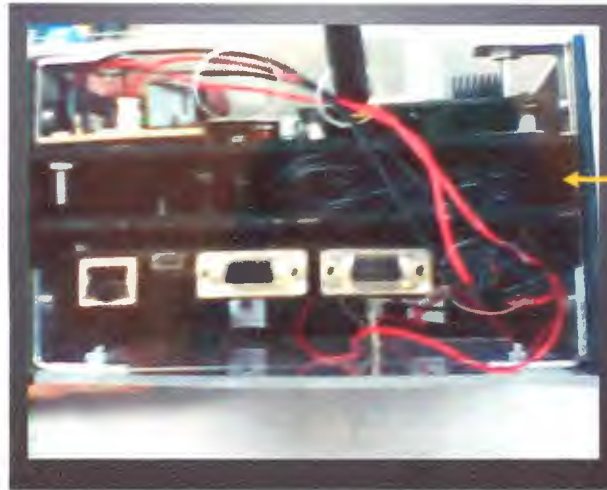
**Parte Posterior 2  
(Rabbit y circuito medidor de rpm)**

Rabbit  
(Serial/Wi-fi)



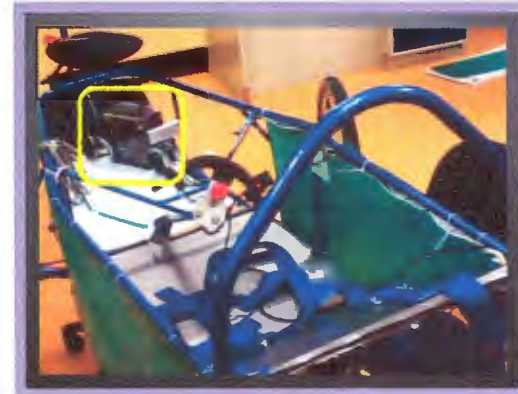
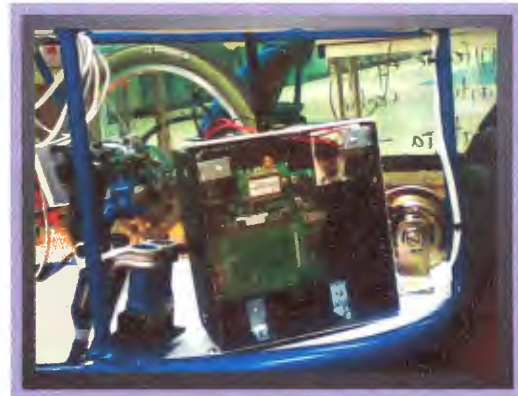
Circuito  
medidor de rpm

### Parte Superior



Baterías

### Circuito montado en el carro



La caja se encuentra en la parte delantera del carro, justo atrás del pedal, de esa forma queda cerca del iPad. Con cables llega la información de manera serial proveniente de los sensores ubicados en la parte trasera del vehículo.

#### 4. 7 Resumen de las pruebas realizadas

Como se puede observar se probaron las distintas etapas que conforman al sistema y se validó su funcionamiento. Los valores enviados por un microcontrolador externo son recibidos correctamente por un puerto serial del microcontrolador de la EVK1100 luego de pasar por un proceso de acoplamiento (compuesto por un dispositivo optoacoplador). El circuito medidor de las revoluciones por minuto también funcionó y entregó los valores al microcontrolador para poder hacer el cálculo de la velocidad.

La etapa que presentó más dificultades fue la comunicación inalámbrica entre la tarjeta de adquisición de datos y el iPad. Los datos se recibían correctamente por el puerto serial del *rabbit* (RS232/Wi-Fi) y se radiaban por *Wi-Fi*. Sin embargo costó trabajo que en todas las ocasiones el iPad recibiera esta información por *Wi-Fi*, la red *Wi-Fi* podía establecerse pero la recepción de datos se dificultaba algunas veces, este punto podría mejorarse realizando un mejor uso del espacio en memoria del iPad.

Para la telemetría se comprobó la escritura de los datos en la tarjeta SD, la lectura de la información en una computadora o laptop y la aplicación web para los diversos dispositivos móviles.

Se pudieron realizar pruebas breves con el sistema montado en el vehículo. Es importante señalar que el sistema funcionó correctamente en un ambiente ruidoso, es decir aún con el motor funcionando y la celda de hidrógeno encendida; la adquisición de datos y la escritura en la tarjeta SD se realizaron adecuadamente.

Sería interesante hacer más pruebas en ambientes diferentes para evaluar aún mejor el sistema completo con el coche en movimiento.



# Capítulo 5. Conclusiones

---

## 5.1 Conclusiones

En conclusión el sistema completo funcionó adecuadamente. A pesar de las dificultades que se presentaron durante el desarrollo, la toma de decisiones para dar solución a cada problema fue pertinente y facilitó la obtención de mejores resultados.

Para tratar de sintetizar el trabajo realizado en el proyecto se enumera una serie de pasos concretos:

- a) Definición de objetivos específicos.
- b) Definición del equipo de instrumentación adecuado para cada etapa.
- c) Almacenamiento de datos en una tarjeta SD.
- d) Aplicación web para despliegue de datos en *pits*.
- e) Aplicación para leer la tarjeta SD en un formato estándar.
- f) Aplicación para recepción de datos por *Wi-Fi* en el iPad.
- g) Circuito medidor de rpm.
- h) Comunicación con la etapa de control.
- i) Acoplamiento del sistema completo con el vehículo

Con el desarrollo del proyecto se logró una mayor experiencia en el uso de diversas tecnologías de la información, tecnologías de comunicación inalámbrica (*bluetooth* y *Wi-Fi*) y uso de varios lenguajes de programación para aplicaciones diferentes.

Algunas de las conclusiones finales que obtuvimos luego de realizar las pruebas y tener dicha experiencia de un proyecto de ingeniería, fueron las siguientes:

- Es necesario realizar la transmisión serial de una manera aislada del resto del sistema.
- Es indispensable la implementación de algoritmos de sincronización en la comunicación de sistemas independientes. En nuestro caso se definió una trama para la recepción de datos de manera serial provenientes de la etapa de control, la cual realizó otro equipo.
- La tecnología *Wi-Fi* satisfizo las necesidades de nuestro proyecto en cuanto a la comunicación entre un módulo y el iPad.
- Se debe prestar atención al tiempo de ejecución de las instrucciones del microcontrolador al usar varios puertos de comunicación.

En general los resultados que obtuvimos los consideramos óptimos en cuanto al funcionamiento del sistema y se cumplió el objetivo inicial, ya que desarrollamos un sistema de adquisición de datos y de telemetría. Sin embargo, es necesario realizar más pruebas reales dentro del vehículo.

## 5.2 Trabajo a futuro

Siempre es bueno hacer una propuesta de las posibles mejoras que se le pueden implementar al proyecto.

Las propuestas que hacemos para mejorar nuestro proyecto son las siguientes:

- Por falta de tiempo e imprevistos no fue posible realizar todas las pruebas deseadas con el sistema dentro del vehículo para la obtención de datos reales. Además es necesario hacer las mismas pruebas pero en ambientes ruidosos y medir el desempeño del sistema.
- Realizar el procesamiento de datos en el servidor. Para el proyecto, los dispositivos móviles situados en *pits* reciben los datos y cada uno hace el procesamiento necesario para poder desplegar las gráficas. Si se tuviera una cantidad mucho mayor de datos, esta solución no sería la más pertinente, lo adecuado sería hacer el procesamiento necesario directamente en el servidor.
- Sin duda lo que le daría un desempeño mejor al sistema sería poder hacer la telemetría en tiempo real, en lugar de almacenar los datos y posteriormente leerlos. Poder enviar los datos hacia *pits* en tiempo real desde el vehículo andando sería de suma relevancia para garantizar un mejor papel en las carreras de autos.
- En el capítulo dedicado al diseño del sistema se hace mención de un sistema de alarmas difusas, es decir que se analicen los datos obtenidos y realice una comparación para lograr mayor eficiencia. El sistema de alarmas ya se explicó en este documento, está realizada la lógica y las simulaciones, sin embargo por cuestiones de tiempo no fue posible implementarlo en el microcontrolador de la EVK1100, así que una mejor al proyecto consistiría en hacer dicha implementación y realizar las pruebas.
- Otro aspecto que se podría lograr para un futuro es hacer un despliegue, en el iPad situado en el coche, con un mapa de la pista en el que se indique la ubicación actual del vehículo dentro de ella y el número de vueltas que ha recorrido.
- También sería interesante poder visualizar la ruta más óptima y poder ir haciendo comparaciones en tiempo real de la ruta que se tiene con su nivel de eficiencia contra el que debería tener.
- Un último aspecto importante es poder realizar la miniaturización del sistema, es decir hacer un circuito que únicamente contenga los dispositivos y elementos necesarios para el funcionamiento del sistema.

## Bibliografía y referencias

### Bibliografía

[11] *Introduction to Wi-Fi Manual del Producto*, Part Number 019-0170 • 090409–B • Printed in U.S.A. Digi International Inc. © 2007-2008

[14] *Hoja de Especificaciones del UC3A0512* (2009). AVR <sup>®</sup> 32, 32-Bit Microcontroller. Atmel Corporation

[17] MiniCore RCM5600W, *Manual de Usuario* (U.S.A., 2009), Digi International Inc.

[22] Sedra/Smith, *Circuitos microelectrónicos*. 4 Ed. Oxford. University press. México. 2001

[24] Valvano W. Jonathan, *Embedded Microcomputer Systems: Real Time Interfacing*, Ed. Brooks/Cole Thomson Learning

[25] C. Schmelkes, *Manual para la presentación de anteproyectos e informes de investigación (tesis)*. México: Oxford, 1998. ISBN 970-613-354-2.

### Referencias electrónicas

[1] Shell Eco-Marathon, extraído de [www.shell.com/home/content/ecomarathon](http://www.shell.com/home/content/ecomarathon) el 20 de agosto 2011.

[2] Rules. Shell Eco-marathon Americas<sup>®</sup> Chapter II 2011 extraído de [http://www.shell.com/home/content/ecomarathon/for\\_participants/rules/](http://www.shell.com/home/content/ecomarathon/for_participants/rules/) el 20 de agosto 2011

[3] Shell Eco-Marathon: EcoCustom Team, <http://www.ecocustom.com/equipo/objetivos.php> 13 de septiembre, 2010.

[4] Shell Eco-Marathon: Synergy team © Copyright 2009. All rights reserved. Designed by Faraz Nawaz <http://www.team-synergy.uet-mct.com/team.html> Septiembre, 2010.

[5] Team Crocodile (2006), Shell Eco-marathon, <http://www-croc.phy.cam.ac.uk/people.php> 11 de septiembre, 2010.

[6] Coventry University's team website, <http://www.cuae.co.uk/Welcome.html> 10 de septiembre, 2010.

[7] Electrón ccm, facebook del equipo, <http://www.facebook.com/pages/Escuderia-Borregos-CCM/91098429879?sk=info>

[8] Spacelink (2007) Satellite Services, extraída de <http://www.spacelinkngt.com/> el 20 de septiembre 2010



[9] Telemetría. KMT - Messtechnik & Telemetrie (2010), <http://www.kmt-gmbh.com/neu/company.php> 20 de septiembre, 2010.

[10] Definición de Wi-Fi extraída de <http://www.misrespuestas.com/que-es-wifi.html> el 25 de marzo 2011.

[12] RS232 Data Interface, Tutorial on Data Interface and Cables. Extraído de <http://www.arcelect.com/rs232.htm>

[13] Diagrama de pines del serial. Imagen extraída de <http://arduino.cc/es/Tutorial/ArduinoSoftwareRS232>

[15] Características de la tarjeta SD. SD Card Association, extraída de <http://www.sdcard.org/developers/about/>

[16] Tabla de clases y velocidades de la tarjeta SD, SD Association (2010) extraída de <https://www.sdcard.org/consumers/speed/> en febrero 2011

[18] Características del iPad. <http://www.apple.com/es/ipad/>

[19] Imagen de la rueda, extraída de <http://www.lacuenca.es/images/rueda3.jpg&imgrefurl=http://www.lacuenca.es/elcampo/agricultura/index.php&usq>

[20] Imagen del sensor extraída de <http://www.pingprobots.com/Sensors.html&usq>

[21] Gráfica del Schmitt Trigger, extraída de <http://www.elektronik-kompendium.de/sites/bau/0209241.htm>

## **Anexos**

## Anexos

### Simulación del funcionamiento de alarmas en Matlab

```
function N = Fuzzy(rpmIN, powerIN)
% Calculates the fuzzification, degree of membership of the received
% parameters and returns the output crisp N.

%fuzzification espace constants
RPM_LEFT = -20;
RPM_RIGHT = 100;
RPM_BEST = 320;
POWER_LEFT = -180;
POWER_RIGHT = 100;
POWER_BEST = 940;

%init values of rpm membership function
rapido = 0;
okRPM = 0;
lento = 0;

%init values of power membership function
mucha = 0;
okP = 0;
poca = 0;

%Calculation of the errors
E_rpm = RPM_BEST - rpmIN;
E_power = POWER_BEST - powerIN;

%calculate RPM membership
if E_rpm <= RPM_LEFT
    rapido = 255;
    okRPM = 0;
    lento = 0;
elseif E_rpm < 0
    rapido = (255*E_rpm)/RPM_LEFT;
    okRPM = 255 - rapido;
    lento = 0;
elseif E_rpm < RPM_RIGHT
    rapido = 0;
    lento = (255*E_rpm)/RPM_RIGHT;
    okRPM = 255 - lento;
else
    rapido = 0;
    okRPM = 0;
    lento = 255;
end
```

## Sistema de Adquisición de Datos y Telemetría para un Kart

```
%calculate POWER membership
if E_power <= POWER_LEFT
    mucha = 255;
    okP = 0;
    poca = 0;
elseif E_power < 0
    mucha = (255*E_power)/POWER_LEFT;
    okP = 255 - mucha;
    poca = 0;
elseif E_power < POWER_RIGHT
    mucha = 0;
    poca = (255*E_power)/POWER_RIGHT;
    okP = 255 - poca;
else
    mucha = 0;
    okP = 0;
    poca = 255;
end

%Calculate Rules:
sube = min(lento, poca);
manten = min(okRPM, okP);
procuraBajar = min(rapido, okP);
intentaSubir = max(min(lento, okP), min(okRPM, poca));
baja = max(min(okRPM, mucha), min(rapido, mucha));
%Calculate the output crisp:
Nbaja = -88;
NprocuraBajar = -48;
Nmanten = 0;
NintentaSubir = 48;
Nsube = 48;

N = (baja*Nbaja + procuraBajar*NprocuraBajar + manten*Nmanten +
intentaSubir*NintentaSubir + sube*Nsube)/(sube + manten + procuraBajar + intentaSubir + baja);

end

function y = FuzzyRPM(x, power)
y = [];
for i=1 : numel(x)
    y = [y Fuzzy(x(i),power)];
end
end

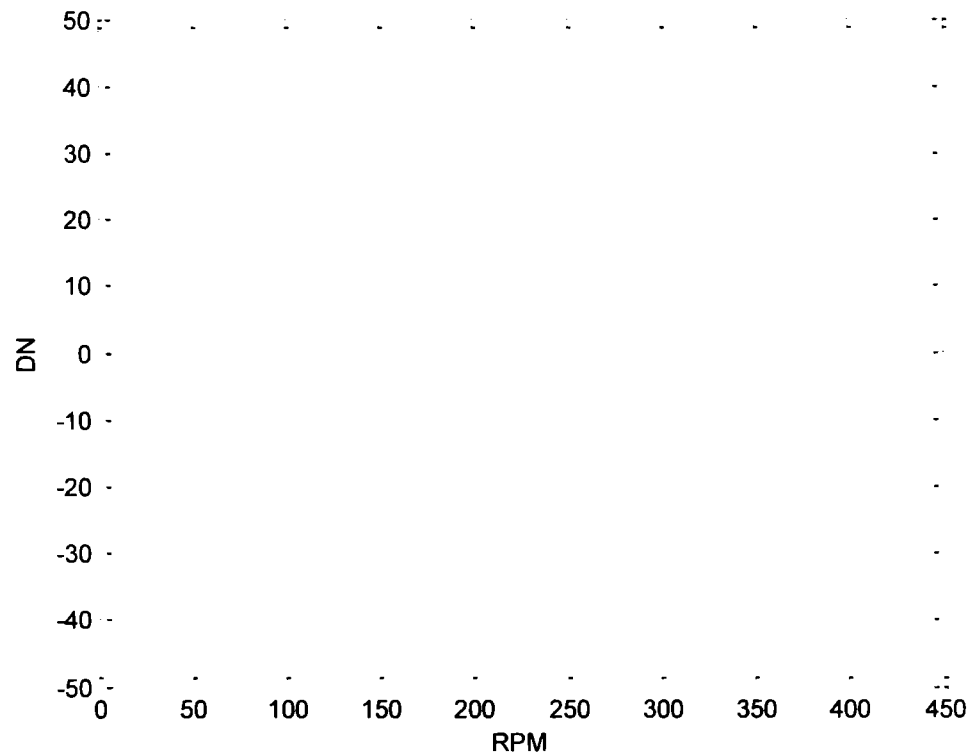
function y = FuzzyPOWER(rpm, x)
y = [];
for i=1 : numel(x)
```

```
    y = [y Fuzzy(rpm,x(i))];  
end  
end  
  
function y = FuzzyBOTH(h, x)  
y = [];  
for i=1 : numel(h)  
    y = [y Fuzzy(h(i),x(i))];  
end  
end
```

Las instrucciones utilizadas se muestran a continuación:

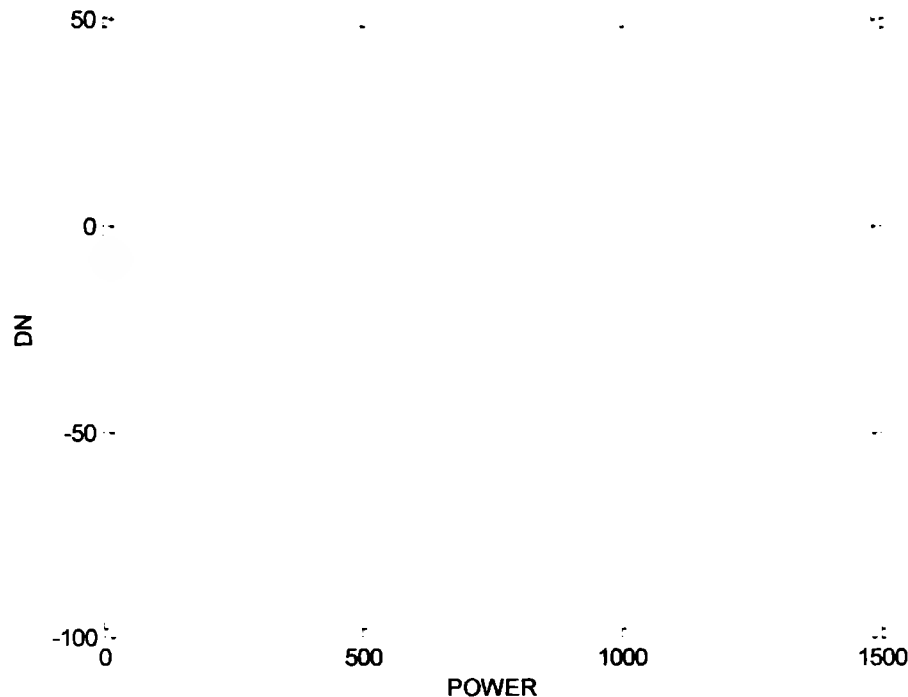
```
>> rpm = 0:5:425;  
>> potencia = 0:5:1500;  
>> y1 = FuzzyRPM(rpm,940);  
>> y2 = FuzzyPOWER(320, potencia);  
>> y3 = FuzzyBOTH(rpm,potencia);
```

Las gráficas obtenidas se muestran a continuación:



Gráfica de  $\Delta N$  contra RPM, mientras la potencia permanece en el valor deseado.

Se puede apreciar que el comportamiento de  $\Delta N$  al ir incrementando las revoluciones por minuto, considerando que el valor de potencia permanece en el deseado. En el rango de 0 a 220 revoluciones por minuto, el sistema le sugerirá al piloto que intente subir la velocidad. Este caso es muy poco probable que ocurra en la práctica, ya que no es posible que el motor esté entregando su potencia eficiente sin que existan revoluciones por minuto cercanas a las óptimas; por lo tanto el sistema considera pertinente sugerir el intento de subir la velocidad. A medida que las revoluciones por minuto se acercan al valor óptimo, el sistema tenderá a sugerirle al piloto que mantenga su velocidad actual. A medida que las revoluciones por minuto sobrepasen el valor óptimo, el sistema le sugerirá al piloto que procure bajar la velocidad.

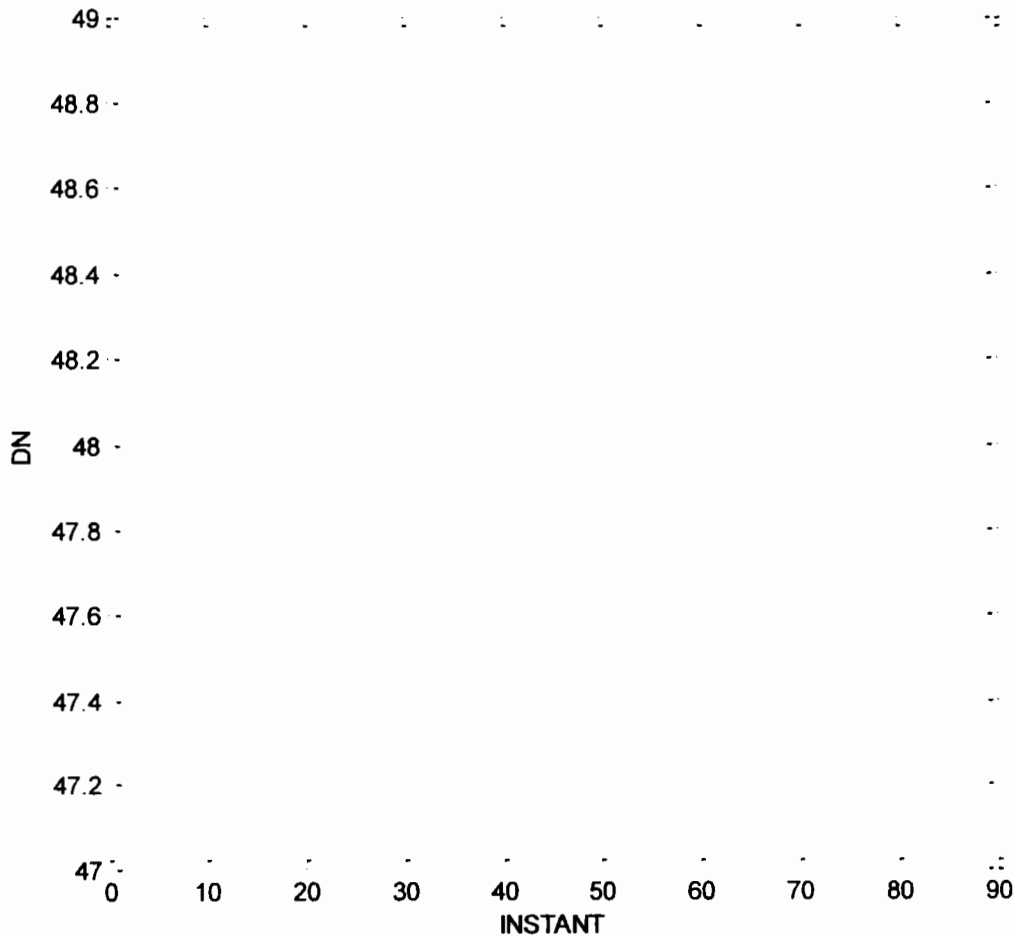


Gráfica de  $\Delta N$  contra potencias, mientras las revoluciones por minuto permanecen en el valor deseado.

Nuevamente podemos observar que el comportamiento de  $\Delta N$ , cuando se varía la potencia entregada por el motor, considerando a las revoluciones por minuto en el valor deseado de manera constante. Nótese el periodo entre 0 y antes de los 1000 Watts, el cual nuevamente es altamente improbable de ocurrir en la realidad. Durante este periodo, el sistema sugerirá al piloto que intente subir la velocidad, tratando de conseguir la potencia deseada, siendo ésta 940 Watts en este caso. Conforme el valor de la potencia se acerque al deseado, el sistema tenderá a sugerirle al piloto que mantenga su velocidad, y tenderá a sugerirle que la disminuya conforme la potencia se aleje por encima del valor deseado. Luego de sobrepasar el valor máximo permitido

dentro del rango de operación, el sistema tenderá a sugerirle al piloto que es necesario bajar la velocidad.

Cabe mencionar que el sistema no le indicará al piloto que es necesario disminuir la velocidad si las revoluciones por minuto sobrepasan las deseadas, si la potencia entregada por el motor se encuentra en el rango de operación permitido, puesto que ello no afectaría ningún componente del sistema; mientras que si se sobrepasa el máximo valor de operación permitido para la potencia entregada por el motor, los resultados pueden ser fatales para el sistema.



Gráfica de  $\Delta N$  obtenido a partir del barrido tanto de rpm como de potencia.

Nótese el comportamiento constante de  $\Delta N$  justo a las 48 unidades. Esto provocaría que el sistema le sugiera constantemente al piloto que intente subir la velocidad. Esta sugerencia no es imperativa, puesto que el sistema notaría que tanto la potencia como las revoluciones por minuto van aumentando.

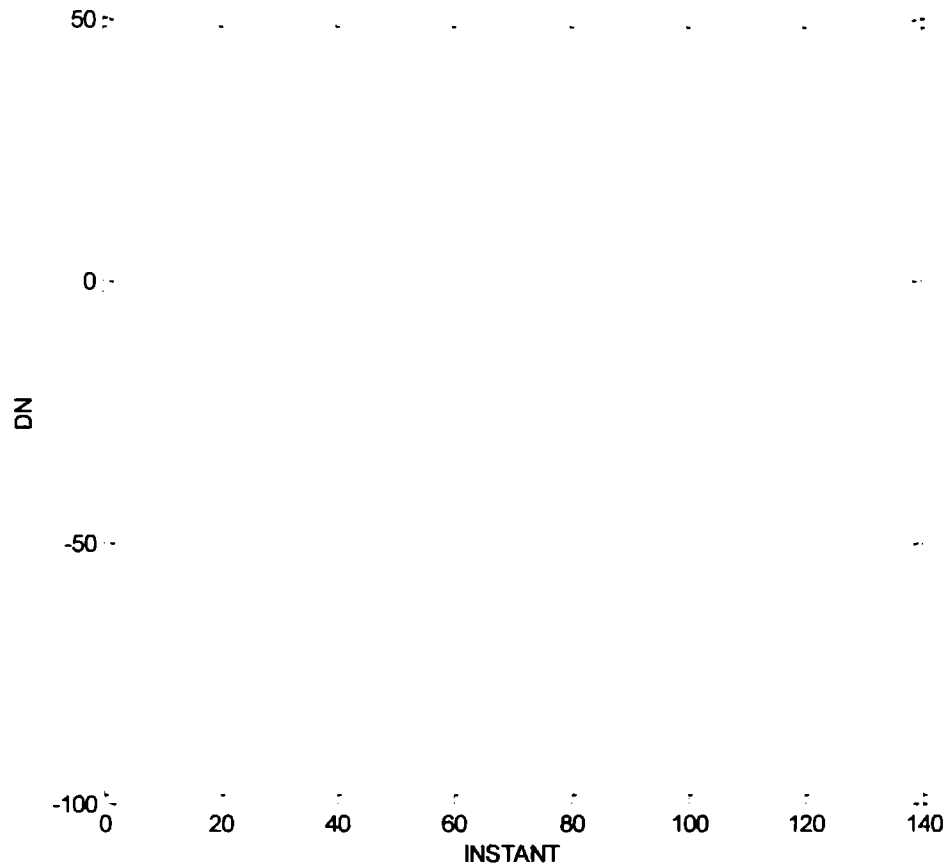
Se decidió hacer una prueba más en Matlab para el barrido simultáneo de revoluciones por minuto y de potencias, esta vez comenzando desde el valor mínimo dentro del rango de operación

hasta sobrepasar el valor máximo de operación en ambos casos. El resultado obtenido se muestra a continuación:

```
>> potencia = 840:5:1500;
```

```
>> rpm = 320:5:980;
```

```
>> y4 = FuzzyBOTH(rpm,potencia);
```



Gráfica de  $\Delta N$  obtenido a partir del barrido tanto de rpm como de potencia, cuando estos valores comienzan en su punto mínimo de operación y sobrepasan el punto máximo de operación permitido. El comportamiento comienza con el sistema sugiriéndole al piloto que intente subir la velocidad. Llega el punto en el que se encuentran cercanos al punto deseado y lo comienzan a sobrepasar. En ese momento, el sistema sugerirá al piloto que procure bajar su velocidad (sección de los -48 constantes). Cuando los valores comienzan a acercarse a los máximos permitidos en el rango de operación, el sistema tenderá a sugerirle al piloto que es necesario bajar la velocidad. Cuando los valores máximos permitidos son sobrepasados, el sistema le indicará al piloto que es necesario bajar la velocidad (estado estacionario en -88).



### Código de Matlab

```

function N = Fuzzy(rpmIN, powerIN)
    Calculates the fuzzification, degree of membership of the received
    parameters and returns the output crisp N.

%fuzzification espace constants
    RPM_LEFT = -20;
    RPM_RIGHT = 100;
    RPM_BEST = 320;
    POWER_LEFT = -180;
    POWER_RIGHT = 100;
    POWER_BEST = 940;

%init values of rpm membership function
    rapido = 0;
    okRPM = 0;
    lento = 0;

%init values of power membership function
    mucha = 0;
    okP = 0;
    poca = 0;

%Calculation of the errors
    E_rpm = RPM_BEST - rpmIN;
    E_power = POWER_BEST - powerIN;

%calculate RPM membership
    if E_rpm <= RPM_LEFT
        rapido = 255;
        okRPM = 0;
        lento = 0;
    elseif E_rpm < 0
        rapido = (255*E_rpm)/RPM_LEFT;
        okRPM = 255 - rapido;
        lento = 0;
    elseif E_rpm < RPM_RIGHT
        rapido = 0;
        lento = (255*E_rpm)/RPM_RIGHT;
        okRPM = 255 - lento;
    else
        rapido = 0;
        okRPM = 0;
        lento = 255;
    end

%calculate POWER membership
    if E_power <= POWER_LEFT
        mucha = 255;
        okP = 0;
        poca = 0;
    elseif E_power < 0
        mucha = (255*E_power)/POWER_LEFT;
        okP = 255 - mucha;
        poca = 0;
    elseif E_power < POWER_RIGHT
        mucha = 0;

```

## Sistema de Adquisición de Datos y Telemetría para un Kart

```
        poca = (255*E_power)/POWER_RIGHT;
        okP = 255 - poca;
    else
        mucha = 0;
        okP = 0;
        poca = 255;
    end

%Calculate Rules:
    sube = min(lento, poca);
    manten = min(okRPM, okP);
    procuraBajar = min(rapido, okP);
    intentaSubir = max(min(lento, okP), min(okRPM, poca));
    baja = max(min(okRPM, mucha), min(rapido, mucha));

%Calculate the output crisp:
    Nbaja = -88;
    NprocuraBajar = -48;
    Nmanten = 0;
    NintentaSubir = 48;
    Nsube = 48;

    N = (baja*Nbaja + procuraBajar*NprocuraBajar + manten*Nmanten +
    intentaSubir*NintentaSubir + sube*Nsube)/(sube + manten + procuraBajar +
    intentaSubir + baja);

end

function y = FuzzyRPM(x, power)
y = [];
for i=1 : numel(x)
    y = [y Fuzzy(x(i),power)];
end
end

function y = FuzzyPOWER(rpm, x)
y = [];
for i=1 : numel(x)
    y = [y Fuzzy(rpm,x(i))];
end
end

function y = FuzzyBOTH(h, x)
y = [];
for i=1 : numel(h)
    y = [y Fuzzy(h(i),x(i))];
end
end
```

### Código del microcontrolador externo

Para poder simular la adquisición de datos provenientes de la etapa de control y poder validar el correcto funcionamiento del sistema, implementamos un programa sobre un microcontrolador ATmega16 hecho en CodeWizardAVR V2.05.0.

El programa se encarga de estar enviando de manera constante los datos necesarios para verificar el estado del vehículo.

```
/******
```

```
This program was produced by the  
CodeWizardAVR V2.05.0 Evaluation  
Automatic Program Generator  
© Copyright 1998-2010 Pavel Haiduc, HP InfoTech s.r.l.  
http://www.hpinfotech.com
```

```
Project :  
Version :  
Date   : 29/03/2011  
Author : Freeware, for evaluation and non-commercial use only  
Company :  
Comments:
```

```
Chip type      : ATmega16  
Program type   : Application  
AVR Core Clock frequency: 8.000000 MHz  
Memory model   : Small  
External RAM size : 0  
Data Stack size : 256
```

```
*****/
```

```
#include <mega16.h>
```

```
// Standard Input/Output functions
```

```
#include <stdio.h>
```

```
#include <delay.h>
```

```
// Declare your global variables here
```

```
void main(void)
```

```
{  
{
```

```
// Input/Output Ports initialization
```

```
// Port A initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
PORTA=0x00;
DDRA=0x00;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0x00;

// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0x00;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=0xFF
// OCO output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer1 Stopped
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
```

```
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: Off
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 2400
UCSRA=0x00;
UCSRB=0x08;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0xCF;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC disabled
ADCSRA=0x00;

// SPI initialization
```

```
// SPI disabled
SPCR=0x00;

// TWI initialization
// TWI disabled
TWCR=0x00;
}
while (1)
{
    delay_ms(20);
    putchar(0xAA);
    putchar(0x30);
    putchar(0x31);
    putchar(0xFF);
    putchar(0x33);
    putchar(0xFF);
    putchar(0x35);
    putchar(0xFF);
    putchar(0x55);
}
}
```