

Universidad Tecnológica Nacional

Proyecto Final

Gestión de flota y control de conductas de
manejo por medio de variables de la ECU

Autor:

- Leiva, Cristian Jesús Leonel

*Proyecto final presentado para cumplimentar los requisitos académicos
para acceder al título de Ingeniero Electrónico*

en la

Facultad Regional Paraná

Octubre 2017

Declaración de autoría:

Yo declaro que el Proyecto Final “Gestión de flota y control de conductas de manejo por medio de variables de la ECU” y el trabajo realizado son propios. Declaro:

- Este trabajo fue realizado en su totalidad, o principalmente, para acceder al título de grado de Ingeniero Electrónico, en la Universidad Tecnológica Nacional, Regional Paraná.
- Se establece claramente que el desarrollo realizado y el informe que lo acompaña no han sido previamente utilizados para acceder a otro título de grado o pre-grado.
- Siempre que se ha utilizado trabajo de otros autores, el mismo ha sido correctamente citado. El resto del trabajo es de autoría propia.
- Se ha indicado y agradecido correctamente a todos aquellos que han colaborado con el presente trabajo.
- Cuando el trabajo forma parte de un trabajo de mayores dimensiones donde han participado otras personas, se ha indicado claramente el alcance del trabajo realizado.

Firmas:

Fecha:

Agradecimientos

Quiero agradecer primeramente a mi familia, quien me ha dado la posibilidad de estudiar y siempre me formo con la mentalidad que es ese el camino para un futuro mejor, que solo se logra con esfuerzo y dedicación.

Agradecer en segunda instancia a mis amigos de la facultad quienes siempre hicieron el camino más fácil ya sea ayudando en lo que sea necesario o con la simpleza de brindar la amistad en el entorno de estudio para hacer que todos estos años sean más fáciles de llevar.

Y por último a toda la facultad, incluyendo personal docente como no docente que siempre cumplieron con su labor ayudando al estudiante para que logre el objetivo que todos deseamos al ingresar a una carrera, que es el simple hecho de ser un gran profesional.

Universidad Tecnológica Nacional

Abstract

Facultad Regional Paraná

Ingeniero en Electrónica

Gestión de flota y control de conductas de manejo por medio de las variables de la ECU

Leiva, Cristian Jesús Leonel

Abstract:

In this project, make a device to position and read variables of the ECU of a vehicle that are transmitted through GPRS to a server capable of displaying by WEB a real-time positioning of the vehicle with the engine parameters of the same.

This was accomplished with five fundamental components, a microcontroller, an integrated OBDII reader called STN1110, a GPRS module the SIM800L, a GPS module Ublox NEO 6M and finally a server where the database and the Web are hosted for the client.

As a result, a real-time tracking and telemetry prototype was obtained capable of providing a competitive solution commercially with large companies but at a reduced cost for those Pymes who want to manage their fleet and control their drivers.

Keywords:

OBDII, GPS, GPRS, DATACENTER, MICROCONTROLLER

Resumen:

En este proyecto, realice un dispositivo de posicionamiento y lectura de variables de la ECU de un vehículo que se transmiten por medio de GPRS a un servidor capaz de mostrar por medio de una WEB el posicionamiento en tiempo real del vehículo con los parámetros del motor del mismo.

Esto lo logré con cinco componentes fundamentales, un microcontrolador, un integrado lector OBDII llamado STN1110, un módulo GPRS el SIM800L, un módulo GPS Ublox NEO 6M y por último un servidor donde se aloja la base de datos y la Web para el cliente. Como resultado obtuve un prototipo de rastreo y telemetría en tiempo real capaz de brindar una solución competitiva comercialmente con las grandes empresas pero con un costo reducido para aquellas pymes que desean gestionar su flota y controlar a sus choferes.

Palabras Clave:

OBDII, GPS, GPRS, SERVIDOR, MICROCONTROLADOR

Índice:

Contenido

Capítulo 1: Introducción	1
Capítulo 2: Desarrollo	2
2.1 Desarrollo general	2
2.1.1 Módulo GPS:.....	4
Estándar NMEA	6
2.1.2 Módulo GPRS:.....	7
2.1.3 Módulo Lector OBDII:	9
Protocolos de comunicación	11
ISO 9141-2 o K-Line:	13
ISO15765-4 /ISO 11898 SAE J2480 o CanBus:.....	13
Modos de Lectura OBDII (PID)	18
2.1.4 Módulo Central:	19
Circuito implementado:.....	20
Problemas y soluciones implementadas.....	22
Análisis del funcionamiento.....	23
Placa Maestra:	23
Placa adquisidora:	24
Desarrollo del software	24
2.2 Diseño Completo.....	26
Capítulo 3: Resultados	30
Capítulo 4: Análisis de Costos	31
Capítulo 5: Discusión y Conclusión.....	33
Capítulo 6: Literatura Citada.....	34

Lista de Figuras:

Figura 1: Esquema General del sistema	2
Figura 2: Esquema general del hardware	3
Figura 3: Esquema general del servidor.	3
Figura 4: Módulo GPS.	4
Figura 5: Esquema del módulo GPS.	5
Figura 6: Módulo GPRS.	8
Figura 7: Esquema del módulo GPRS.	9
Figura 8: Esquema central OBDII.	10
Figura 9: Esquema circuito K-Line.	11
Figura 10: Esquema circuito CanBus.	12
Figura 11: Conector OBDII.	12
Figura 12: Niveles de tensión CanBus.	15
Figura 13: Trama CanBus.	17
Figura 14: Campos Trama CanBus.	17
Figura 15: Formato Trama OBDII.	18
Figura 15: Esquema del microcontrolador.	20
Figura 16: Placa Central.	21
Figura 17: Placa de alimentación.	21
Figura 18: Placa Adquisidora (OBDII).	22
Figura 19: Esquema general del servidor.	25
Figura 20: Recorrido generado sobre archivo KML.	26
Figura 21: Producto final.	27
Figura 22: Placa adquisidora (Intérprete OBDII).	28
Figura 24: Placa Central.	28
Figura 23: Web Final.	29

Lista de Tablas

Tabla 1: Trama GGA.	7
Tabla 2: Trama RMC.	7
Tabla 3: Costos.	31

Lista de Abreviaciones

ECU: Engine Control Unit (Unidad de control de motor)

OBD: On Board Diagnostics (Diagnostico de abordó)

GPS: Global Positioning System (Sistema de posicionamiento global)

GPRS: General Packet Radio Service

PID: Parameter ID (ID de parámetro)

Dedicado a: *Mi familia y amigos...*

Capítulo 1: Introducción

En el negocio de la logística y el transporte, cada vez aparecen más empresas dedicadas a realizar rastreos satelitales con el fin de brindar un servicio de seguridad y/o logístico de posicionamiento para que el cliente pueda monitorear su flota en todo momento, ya sea para reducir costos operativos, controlar su carga, vigilar a sus choferes, gestionar de forma eficaz los recursos que le pertenecen, entre otros. Todas estas empresas se basan en el uso de un equipo de rastreo por GPS con comunicación GPRS o 3G.

Hasta la actualidad los datos como velocidad, rumbo del vehículo, posición, odómetro de viaje, etc... parecían ser suficientes para los clientes, pero en un mundo en el cual la tecnología avanza tan rápido y los vehículos cuentan cada vez con más sensores y componentes electrónicos que permiten controlar los parámetros del motor llevan al usuario a poder hacer un control más exhaustivo sobre su flota como es conocer los datos sobre gases de emisión y conductas de manejo que puedan mejorar el rendimiento de sus vehículos para tener un consumo óptimo con el menor gasto posible, de aquí nace la idea de crear un sistema que controle estos parámetros y los combine con los datos clásicos de posicionamiento de un AVL, si bien equipos de este tipo hoy ya comenzaron a verse a nivel internacional, aun no tienen gran auge en el ámbito nacional lo que lleva a mi tesis la oportunidad de generar una oportunidad de negocio.

Un hardware que no es capaz de mostrar su funcionamiento de forma sencilla y clara para un cliente no es factible comercializarlo, por lo cual se desarrolló en este proyecto final una web capaz de mostrar los parámetros leídos del motor y dar el posicionamiento de la unidad en tiempo real tal como lo realizan las grandes empresas del negocio del rastreo, a diferencia que estas usan una diversidad de equipos AVL diferentes los cuales no todos brindan los mismos datos, de manera que con mi tesis se logra una relación unívoca entre el hardware y el software, logrando homogeneidad en los datos que perciba el cliente.

Capítulo 2: Desarrollo

2.1 Desarrollo general

Presento a continuación, un esquema general del sistema para una mejor comprensión del desarrollo de este proyecto final:



Figura 1: Esquema General del sistema

El hardware desarrollado se encuentra instalado en la camioneta, por medio de la red GPS recibe datos para conocer su ubicación exacta y luego de combinarlo con las variables de la ECU las transmite por medio de GPRS a un DATA CENTER quien es el que aloja los datos de todas las unidades instaladas, una vez que la información se encuentra almacenada, entonces cualquier dispositivo que pueda acceder a la web podrá ver los datos de los vehículos en tiempo real.

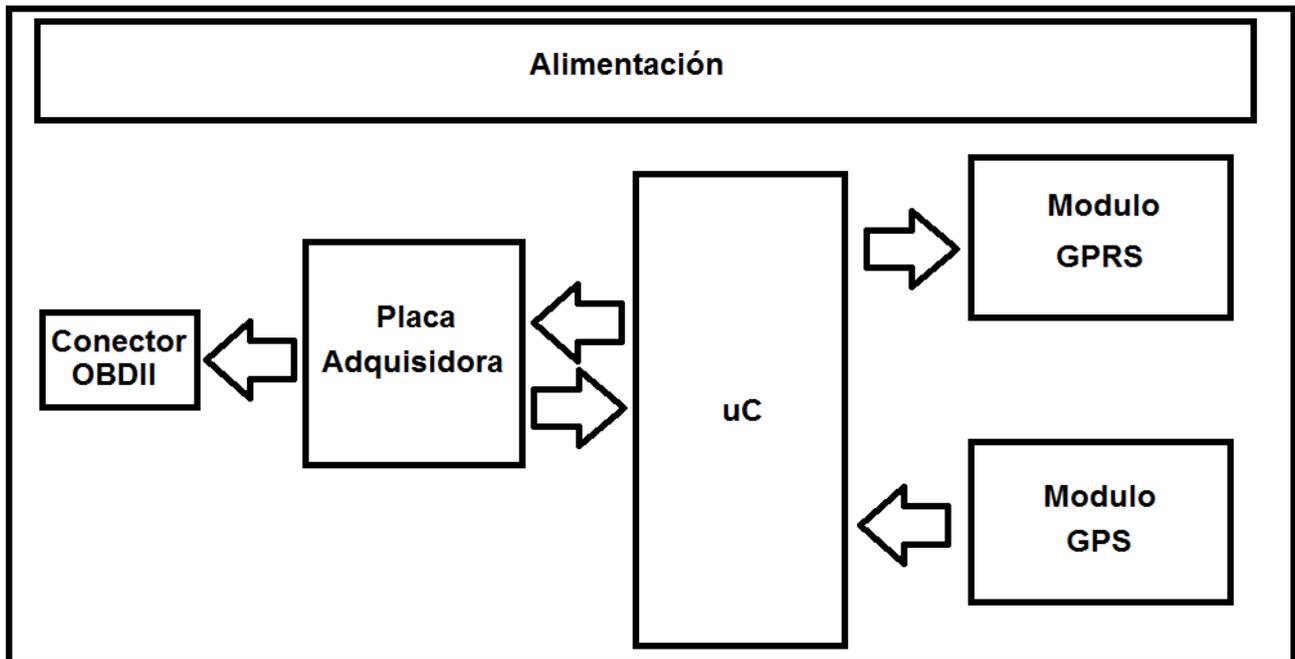


Figura 2: Esquema general del hardware

El esquema general del hardware muestra todas las partes del desarrollo físico, este cuenta de un módulo GPS encargado de brindar los datos de posicionamiento, la placa adquisidora encargada de leer las variables de la ECU del vehículo por medio del conector OBDII, el módulo GPRS que se encarga de enviar los datos al servidor, y una unidad central de control destinada a combinar los datos del GPS con los de la ECU y transmitirlos vía GPRS al servidor.

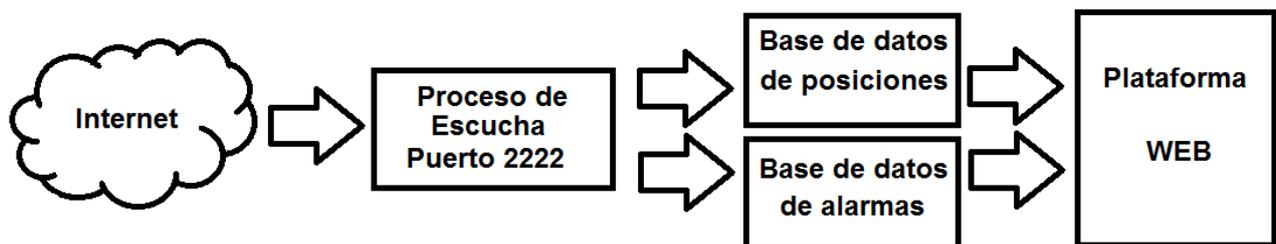


Figura 3: Esquema general del servidor.

Por último, cuento con el servidor que incluye base de datos y web para mostrar las variables tomadas del vehículo con su respectivo posicionamiento.

Desde internet llegan los paquetes dirigidos a una IP y puerto del servidor, en este está corriendo un proceso de escucha del puerto y atrapa las tramas que provienen del hardware, realiza un parseo y la inserta en la base de datos correspondiente. Por último la

web refresca los datos cada cierto tiempo tomando los últimos valores que ingresaron a la base de datos.

2.1.1 Módulo GPS:

- Módulo Ublox NEO-6M
- Comunicación serial
- Alimentación 3,5 a 5 V
- EEPROM para guardar datos de configuración
- Exactitud de 1us
- Tiempo promedio de primer inicio 38 segundos
- Frecuencia receptora L1 (1575.42 Mhz)

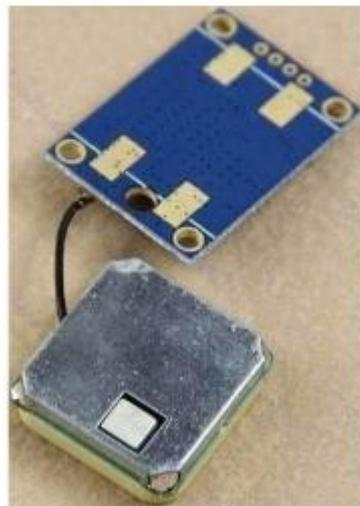


Figura 4: Módulo GPS.

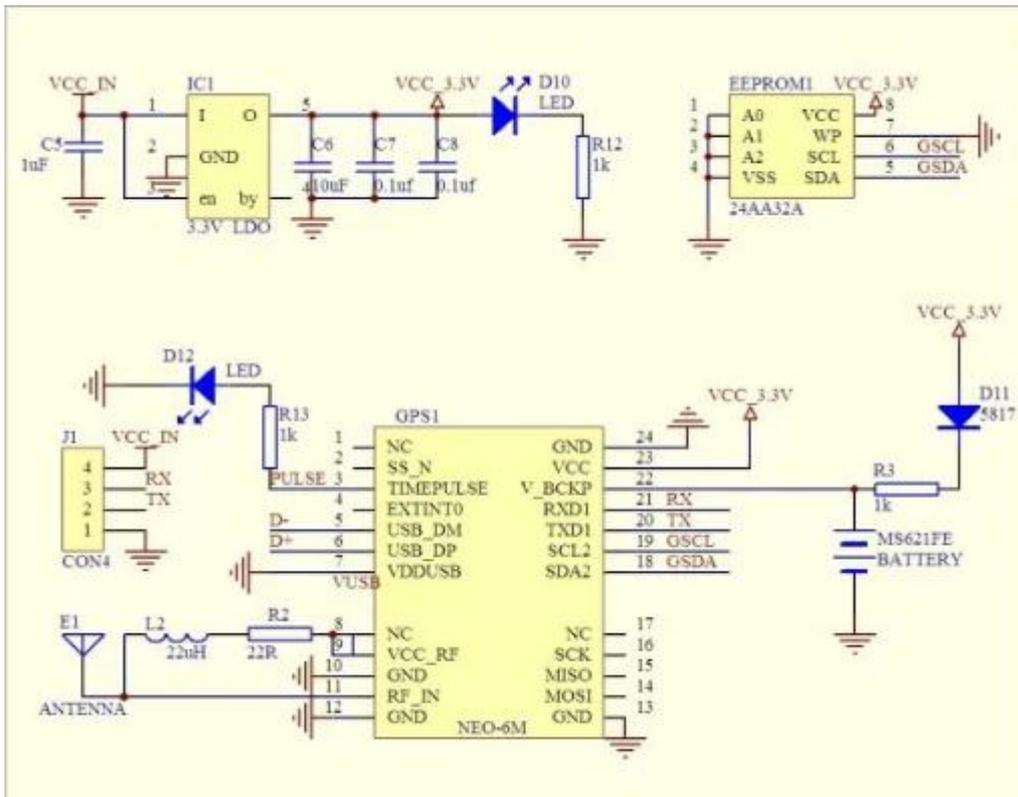


Figura 5: Esquema del módulo GPS.

El Sistema de Posicionamiento Global, más conocido por sus siglas en inglés, GPS (siglas de Global Positioning System), es un sistema que permite determinar en toda la Tierra la posición de un objeto (una persona, un vehículo) con una precisión de hasta centímetros (si se utiliza GPS diferencial), aunque lo habitual son unos pocos metros de precisión. El sistema fue desarrollado, instalado y empleado por el Departamento de Defensa de los Estados Unidos.

El GPS funciona mediante una red de 24 satélites en órbita sobre el planeta Tierra, a 20.200 km de altura, con trayectorias sincronizadas para cubrir toda la superficie de la Tierra. Cuando se desea determinar la posición, el receptor que se utiliza para ello localiza automáticamente como mínimo tres satélites de la red (Posición 3D), de los que recibe unas señales indicando la identificación y la hora del reloj de cada uno de ellos. Con base en estas señales, el aparato sincroniza el reloj del GPS y calcula el tiempo que tardan en llegar las señales al equipo, y de tal modo mide la distancia al satélite mediante el método de trilateración inversa, el cual se basa en determinar la distancia de cada satélite al punto de medición. Conocidas las distancias, se determina fácilmente la propia posición relativa respecto a los satélites. Conociendo además las coordenadas o posición de cada uno de ellos por la señal que emiten, se obtiene la posición absoluta o coordenadas reales del punto de medición. También se consigue una exactitud extrema en el reloj del GPS, similar a la de los relojes atómicos que lleva a bordo cada uno de los satélites.

Cada satélite GPS emite continuamente un mensaje de navegación a 50 bits por segundo en la frecuencia transportadora de microondas de aproximadamente 1.600 MHz. Más

concretamente, todos los satélites emiten a 1575,42 MHz (esta es la señal L1) y 1227,6 MHz (la señal L2).

La señal GPS proporciona la “hora de la semana” precisa de acuerdo con el reloj atómico a bordo del satélite, el número de semana GPS y un informe de estado para el satélite de manera que puede deducirse si es defectuoso. Cada transmisión dura 30 segundos y lleva 1500 bits de datos codificados. Esta pequeña cantidad de datos está codificada con una secuencia pseudoaleatoria (PRN) de alta velocidad que es diferente para cada satélite. Los receptores GPS conocen los códigos PRN de cada satélite y por ello no sólo pueden decodificar la señal sino que la pueden distinguir entre diferentes satélites.

Las transmisiones son cronometradas para empezar de forma precisa en el minuto y en el medio minuto tal como indique el reloj atómico del satélite. La primera parte de la señal GPS indica al receptor la relación entre el reloj del satélite y la hora GPS. La siguiente serie de datos proporciona al receptor información de órbita precisa del satélite.

Estándar NMEA

NMEA 0183 es una especificación combinada eléctrica y de datos entre aparatos electrónicos marinos y, también, más generalmente, receptores GPS.

El protocolo NMEA 0183 es un medio a través del cual los instrumentos marítimos y también la mayoría de los receptores GPS pueden comunicarse los unos con los otros. Ha sido definido, y está controlado, por la organización estadounidense National Marine Electronics Association.

Desde el punto de vista protocolar, el estándar NMEA tiene definidas varias tramas de datos con la información pertinente del posicionamiento del GPS, para el desarrollo de esta tesis solo implementé la lectura de dos de estas tramas, con las cuales conseguí obtener toda la información necesaria para el posicionamiento, las tramas implementadas fueron:

\$GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,*47

Donde:

GPGGA	Cabecera
123519	Hora de posicionamiento (UTC GMT 0)
4807.038,N	Latitud (48° 07.038' N)
01131.000,E	Longitud (11° 31,000' E)
1	Calidad de posicionamiento
08	Cantidad de satelites
0.9	Hdop
545.4, M	Altitud en metros sobre nivel del mar
*47	Checksum

Tabla 1: Trama GGA.

\$GPRMC,123519,A,4807.038,N,01131.000,E,022.4,084.4,230394,003.1,W*6A

Donde:

GPRMC	Cabecera
022.4	Velocidad en nudos
084.4	Rumbo en grados
*47	Checksum

Tabla 2: Trama RMC.

2.1.2 Módulo GPRS:

- Módulo SIM800L GSM/GPRS Cuartibanda
- Alimentación 3.3 a 5 V
- Comunicación Serial
- Configuración Mediante comandos AT



Figura 6: Módulo GPRS.

El servicio general de paquetes vía radio, en inglés: General Packet Radio Service (GPRS), fue creado en la década de los 80. Una conexión GPRS está establecida por la referencia a su nombre de punto de acceso (APN). Con GPRS se pueden utilizar servicios como Wireless Application Protocol (WAP), servicio de mensajes cortos (SMS), Multimedia Messaging System (MMS), Internet y para los servicios de comunicación, como el correo electrónico y la World Wide Web (WWW). Para fijar una conexión de GPRS para un módem inalámbrico, un usuario debe especificar un APN, opcionalmente un nombre y contraseña de usuario, y muy raramente una dirección IP, todo proporcionado por el operador de red. La transferencia de datos de GPRS se cobra por volumen de información transmitida (en kilo o megabytes), mientras que la comunicación de datos a través de conmutación de circuitos tradicionales se factura por minuto de tiempo de conexión, independientemente de si el usuario utiliza toda la capacidad del canal o está en un estado de inactividad. Por este motivo, se considera más adecuada la conexión conmutada para servicios como la voz que requieren un ancho de banda constante durante la transmisión, mientras que los servicios de paquetes como GPRS se orientan al tráfico de datos. La tecnología GPRS como bien lo indica su nombre es un servicio orientado a radio-enlaces que da mejor rendimiento a la conmutación de paquetes en dichos radio-enlaces.

El acceso al canal utilizado en GPRS se basa en divisiones de frecuencia sobre un dúplex y TDMA. Durante la conexión, al usuario se le asigna un canal físico, formado por un bloque temporal en una portadora concreta. Ese canal será de subida o bajada dependiendo de si el usuario va a recibir o enviar datos. Esto se combina con la multiplexación estadística en el dominio del tiempo, permitiendo a varios usuarios

compartir el mismo canal físico, ya sea de subida o de bajada. Los paquetes tienen longitud constante, correspondiente a la ranura de tiempo del GSM. El canal de bajada utiliza una cola de tipo First in, first out (FIFO; primero en entrar, primero en salir) para los paquetes en espera, mientras que el canal de subida utiliza un esquema similar al de ALOHA con reserva.

En resumen, se utiliza un sistema similar al ALOHA ranurado durante la fase de contención, y TDMA con una cola FIFO durante la fase de transmisión de datos.

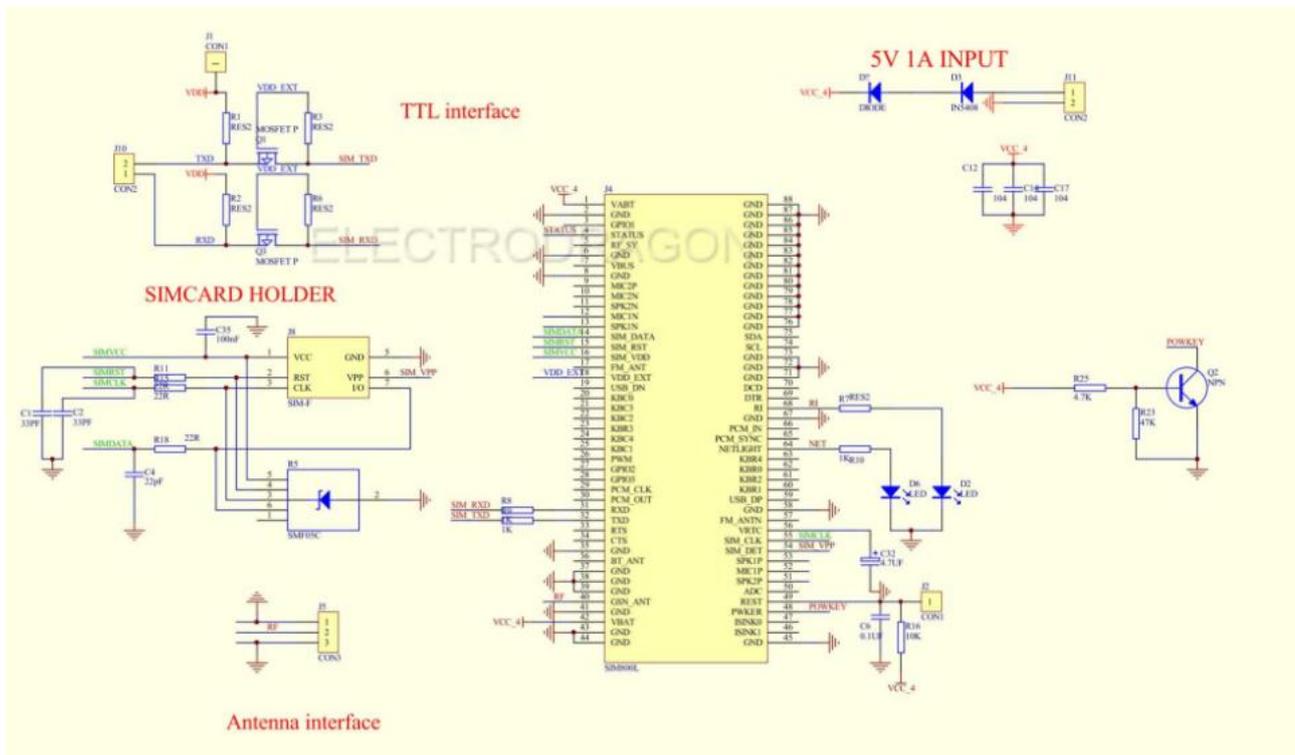


Figura 7: Esquema del módulo GPRS.

1.1.3 Modulo Lector OBDII:

- Realizado con el integrado STN1110
- Compatible con los 5 protocolos OBDII

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

OBD II es la segunda generación del sistema de diagnóstico a bordo, sucesor de OBD I. Alerta al conductor cuando el nivel de las emisiones es 1.5 mayor a las diseñadas. A diferencia de OBD I, OBD II detecta fallos eléctricos, químicos y mecánicos que pueden afectar al nivel de emisiones del vehículo. Por ejemplo, con OBD I, el conductor no se

daría cuenta de un fallo químico del catalizador. Con OBD II, los dos sensores de oxígeno, uno antes y el otro después del catalizador, garantizan el buen estado químico del mismo.

El sistema verifica el estado de todos los sensores involucrados en las emisiones, como por ejemplo la inyección o la entrada de aire al motor. Cuando algo falla, el sistema se encarga automáticamente de informar al conductor encendiendo una luz indicadora de fallo (Malfunction Indication Lamp (MIL), también conocida como Check Engine o Service Engine Soon).

Para ofrecer la máxima información posible para el mecánico, guarda un registro del fallo y las condiciones en que ocurrió. Cada fallo tiene un código asignado. El mecánico puede leer los registros con un dispositivo que envía comandos al sistema OBD II llamados PID (Parameter ID).

Generalmente el conector OBD II suele encontrarse en la zona de los pies del conductor, consola central o debajo del asiento del copiloto.

Actualmente se puede conectar con la máquina de diagnóstico de diferentes maneras, mediante Bluetooth, WiFi, USB, cayendo en desuso el protocolo de conexión, el puerto serie (RS232). Este enlace, unido a un software ejecutándose desde un ordenador o un terminal móvil permite la monitorización en tiempo real de códigos de error y diversos parámetros directamente de la centralita del motor tales como las revoluciones del motor, el consumo de combustible en tiempo real (sin que el automóvil lleve equipado ordenador de abordo) o la temperatura del aceite, entre muchos otros parámetros dependiendo del modelo. El controlador ELM327 es el más extendido para establecer dichos enlaces entre la centralita del motor y el dispositivo con el software instalado.

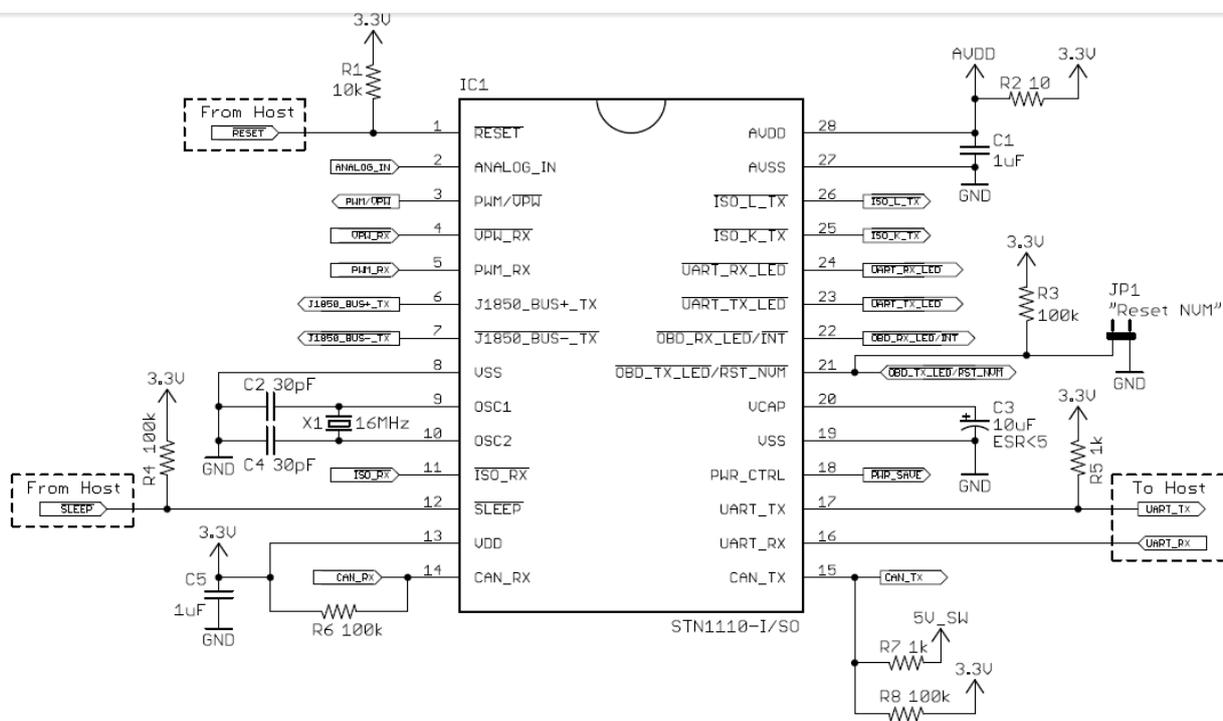


Figura 8: Esquema central OBDII.

Protocolos de comunicación

Existen 5 protocolos de comunicación del sistema OBDII con los lectores de fallas. Los fabricantes han escogido que protocolo utilizar y todos los vehículos que salen de su fábrica salen con el mismo protocolo, por tanto es fácil saber que tipo de protocolo funcionan las comunicaciones de nuestro coche.

- ISO 9141-2 en vehículos Europeos, Asiáticos y Chrysler con variantes (Key Word Protocol = Palabra Clave)
- SAE J1850 VPW que significa Ancho de Pulso Variable (Variable Pulse Width) y lo utiliza GM USA (General Motors)
- SAE J1850 PWM que indica Modulación Ancho de Pulso (Pulse Width Modulation) utilizado por Ford USA.
- KWP 1281 y KWP 2000 utilizado por el grupo VAG.
- ISO15765-4 /ISO 11898 SAE J2480 (CAN) Pin 6 CAN_H i + pin 14 CAN_Low

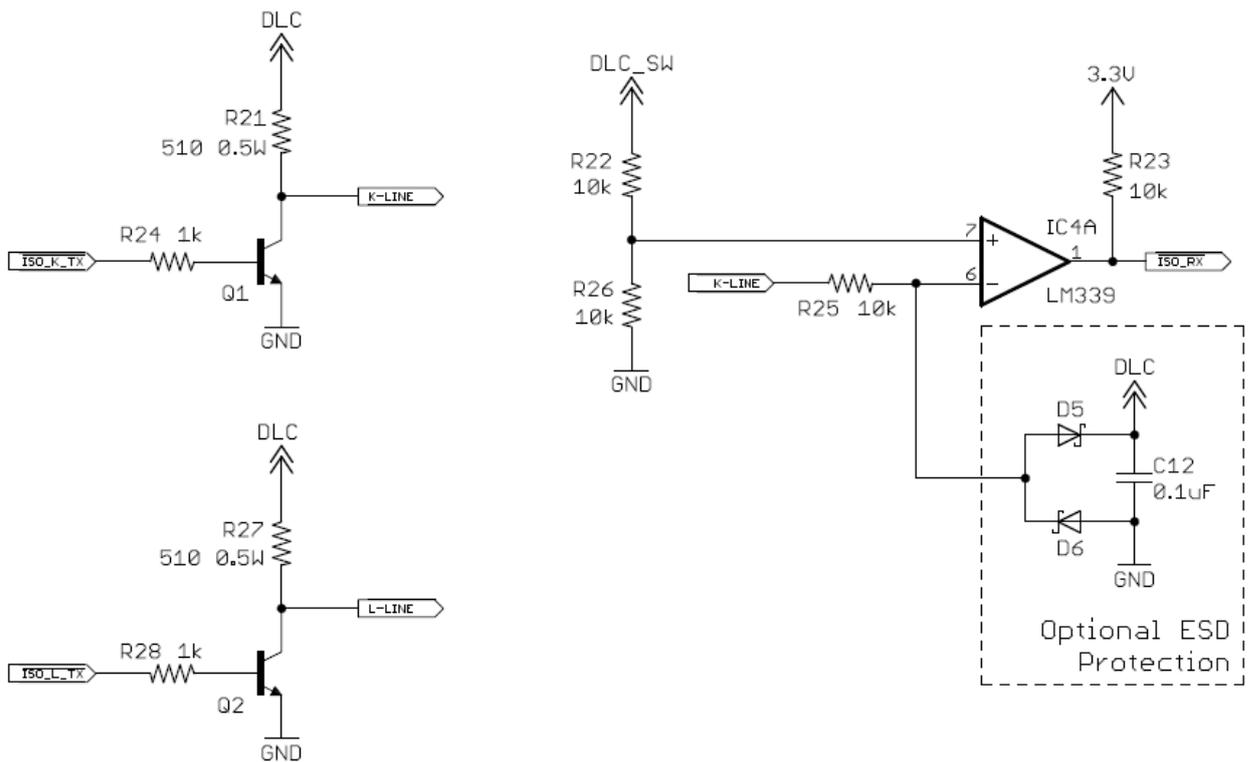


Figura 9: Esquema circuito K-Line.

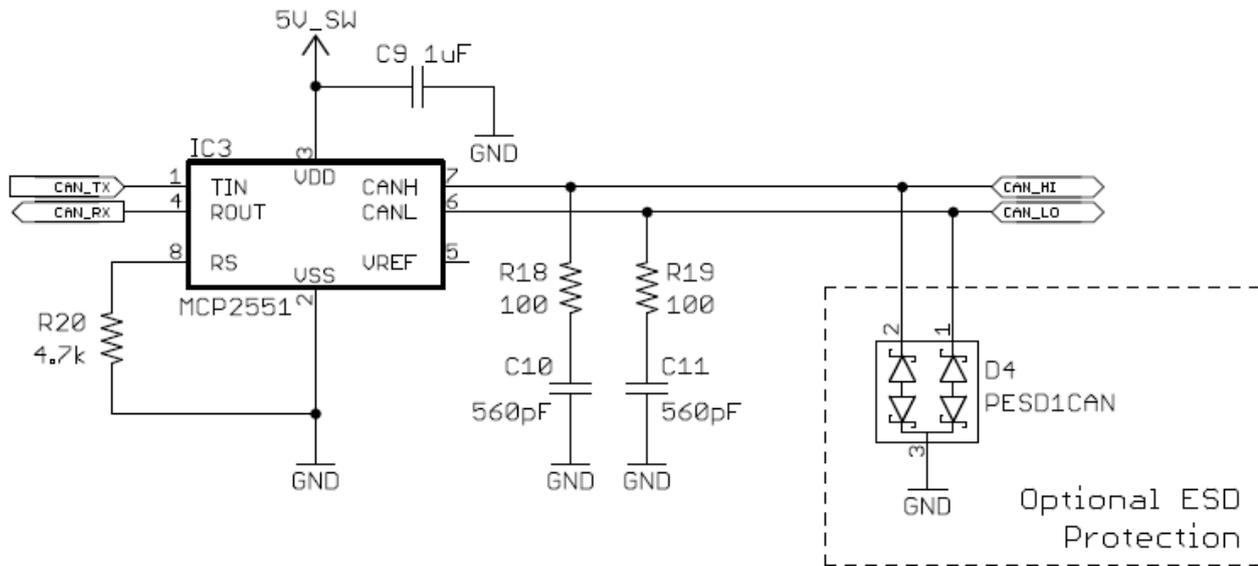
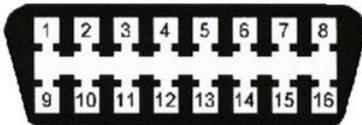


Figura 10: Esquema circuito CanBus.

Como es fácil deducir, cada uno de estos protocolos, requiere de un tratamiento de la información diferente, antes de conectar el OBDII con el PC. Y por tanto, se requieren interfaces de conexión diferentes. Esto no es del todo exacto, ya que existe la posibilidad de fabricar un interfaz de conexión del OBDII con el PC, capaz de utilizar todos los protocolos e incluso seleccionar automáticamente cual es el protocolo utilizado por el vehículo a conectar.

Terminales del Conector OBDII



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

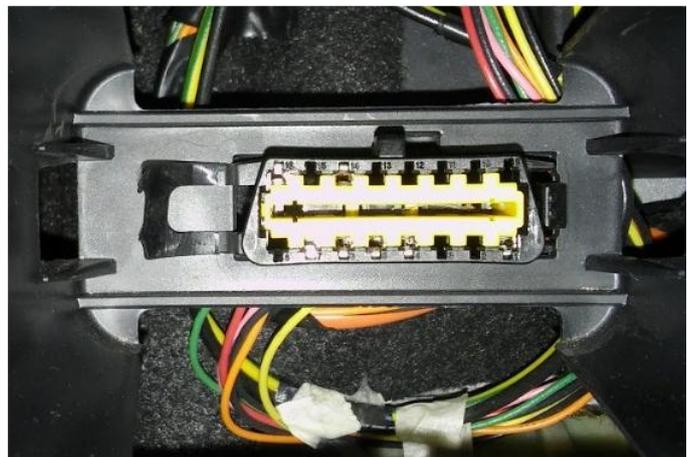


Figura 11: Conector OBDII.

Para el desarrollo de mi tesis, solo implementé 2 protocolos, que son el ISO 9141-2, por ser el que tiene el vehículo en el cual realice las pruebas y el ISO15765-4 por ser el protocolo que se presume que va a ser el estándar a adoptar a nivel global y que hoy en día lo tienen la mayoría de los vehículos pesados.

ISO 9141-2 o K-Line:

La línea K es un tipo de bus monohilo bidireccional descrito por las normas ISO 9141 e ISO 14230-1 que se utiliza en automoción.

Análogamente, la línea L es un cable similar que se utiliza para el establecimiento de la comunicación con una centralita electrónica, lo que se conoce como “fase de excitación”. Tras establecerse el diálogo, la línea L toma permanentemente un valor de señal alto. En la práctica, una inicialización de este tipo es inusual, puesto que en la mayoría de los coches se realiza a través de la propia línea K

Las líneas K y L se utilizan primordialmente para la comunicación con el exterior (por ejemplo, para la diagnosis offboard de centralitas electrónicas en un taller). El flujo de datos en un instante determinado solamente puede discurrir en uno de los dos sentidos. A pesar de eso, se conserva la característica fundamental de posibilitar la comunicación de varios implicados a través de un mismo cable. Las tasas binarias típicas son 9.600 y 10.400 baudios.

El bus trabaja con niveles de señal definidos respecto a la tensión de batería del vehículo (típicamente $V_{bat}=12\text{ V}$ y 24 V). Así, un cero lógico tomará un valor del 0–30% de V_{bat} y un 1 lógico del 70–100% de V_{bat} . El valor exacto de ese porcentaje depende del sentido la comunicación (vehículo a tester de diagnosis o viceversa).

ISO15765-4 /ISO 11898 SAE J2480 o CanBus:

CAN (acrónimo del inglés Controller Area Network) es un protocolo de comunicaciones desarrollado por la firma alemana Robert Bosch GmbH, basado en una topología bus para la transmisión de mensajes en entornos distribuidos. Además ofrece una solución a la gestión de la comunicación entre múltiples CPUs (unidades centrales de proceso).

El protocolo de comunicaciones CAN proporciona los siguientes beneficios:

- Ofrece alta inmunidad a las interferencias, habilidad para el autodiagnóstico y la reparación de errores de datos.
- Es un protocolo de comunicaciones normalizado, con lo que se simplifica y economiza la tarea de comunicar subsistemas de diferentes fabricantes sobre una red común o bus.
- El procesador anfitrión (host) delega la carga de comunicaciones a un periférico inteligente, por lo tanto el procesador anfitrión dispone de mayor tiempo para ejecutar sus propias tareas.
- Al ser una red multiplexada, reduce considerablemente el cableado y elimina las conexiones punto a punto, excepto en los enganches.

Principales características del CanBus

CAN se basa en el modelo productor/consumidor, el cual es un concepto, o paradigma de comunicaciones de datos, que describe una relación entre un productor y uno o más consumidores. CAN es un protocolo orientado a mensajes, es decir la información que se va a intercambiar se descompone en mensajes, a los cuales se les asigna un identificador y se encapsulan en tramas para su transmisión. Cada mensaje tiene un identificador único dentro de la red, con el cual los nodos deciden aceptar o no dicho mensaje. Dentro de sus principales características se encuentran:

- Prioridad de mensajes.
- Garantía de tiempos de latencia.
- Flexibilidad en la configuración.
- Recepción por multidifusión (multicast) con sincronización de tiempos.
- Sistema robusto en cuanto a consistencia de datos.
- Sistema multimaestro.
- Detección y señalización de errores.
- Retransmisión automática de tramas erróneas
- Distinción entre errores temporales y fallas permanentes de los nodos de la red, y desconexión autónoma de nodos defectuosos.

CAN fue desarrollado inicialmente para aplicaciones en los automóviles y por lo tanto la plataforma del protocolo es resultado de las necesidades existentes en el área de la automoción. La Organización Internacional para la Estandarización (ISO, International Organization for Standardization) define dos tipos de redes CAN: una red de alta velocidad (hasta 1 Mbit/s), bajo el estándar ISO 11898-2, destinada para controlar el motor e interconectar las unidades de control electrónico (ECU); y una red de baja velocidad tolerante a fallos (menor o igual a 125 kbit/s), bajo el estándar ISO 11519-2/ISO 11898-3, dedicada a la comunicación de los dispositivos electrónicos internos de un automóvil como son control de puertas, techo corredizo, luces y asientos.

Tipos de CanBus

La especificación de los buses CAN está recogida en el conjunto de estándares ISO 11898. Dicha especificación define las dos primeras capas, la capa física y la capa de enlace de datos, del modelo OSI de interconexión de sistemas. Sobre la base de dichos estándares, los buses CAN se pueden clasificar en dos tipos:

- CAN de alta velocidad (hasta 1 Mbit/s).
- CAN de baja velocidad tolerante a fallos (hasta 125 kbit/s).

Capa Física

Define los aspectos del medio físico para la transmisión de datos entre nodos de una red CAN, las características materiales y eléctricas y la transmisión del flujo de bits a través del bus.

Niveles de tensión del bus CAN

La transmisión de señales en un bus CAN se lleva a cabo a través de dos cables trenzados. Las señales de estos cables se denominan CAN_H (CAN high) y CAN_L (CAN low) respectivamente. El bus tiene dos estados definidos: estado dominante y estado recesivo. En estado recesivo, los dos cables del bus se encuentran al mismo nivel de tensión (common-mode voltage), mientras que en estado dominante hay una diferencia de tensión entre CAN_H y CAN_L de al menos 1,5 V. La transmisión de señales en forma de tensión diferencial, en comparación con la transmisión en forma de tensiones absolutas, proporciona protección frente a interferencias electromagnéticas.

La tensión en modo común puede estar, según la especificación, en cualquier punto entre -2 y 7 V. La tensión diferencial del bus (la diferencia entre CAN_H y CAN_L) en modo dominante debe estar entre 1,5 y 3 V. No se especifica, en cambio, que la tensión de modo común, cuando el bus está en modo recesivo, deba estar comprendida entre la tensión de CAN_L y la tensión de CAN_H cuando el bus está en modo dominante. Esto permite la conexión directa entre nodos que operen a distintas tensiones, e incluso nodos que sufran diferencia de tensión entre sus respectivas tierras.

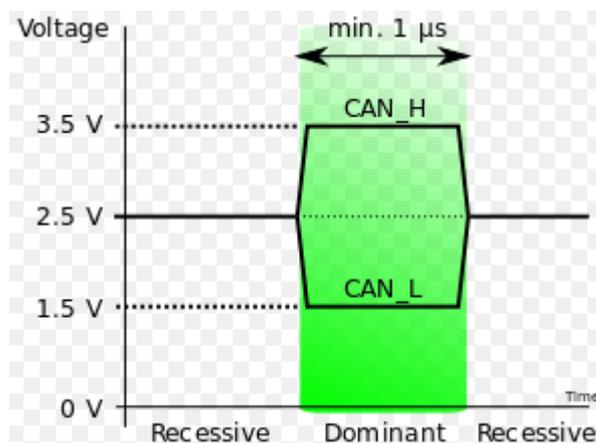


Figura 12: Niveles de tensión CanBus.

Capa de enlace de datos

El protocolo CAN proporciona un acceso multimaestro al bus con una resolución determinista de las colisiones. La capa de enlace de datos define el método de acceso al medio así como los tipos de tramas para el envío de mensajes.

Acceso al Medio

La especificación del CAN usa los términos “dominante” y “recesivo” para referirse a los bits, donde un bit dominante equivale al valor lógico 0 y un bit recesivo equivale al valor lógico 1. El estado inactivo del bus es el estado recesivo (valor lógico 1). Cuando dos nodos intentan transmitir bits diferentes se denomina colisión y el valor del bit dominante prevalece sobre el valor del bit recesivo. En ese caso el nodo que intentaba transmitir el valor recesivo detecta la colisión y pasa a modo pasivo, es decir, deja de transmitir para escuchar lo que transmite el otro nodo. Por esta razón es importante que todos los nodos estén sincronizados y muestreen todos los bits del bus simultáneamente.

El arbitraje se produce durante los primeros bits de una trama o mensaje, durante la transmisión de lo que se conoce como identificador del mensaje. Al final del proceso de arbitraje sólo debe quedar un nodo con el control del bus. Por ello cada nodo debe manejar identificadores únicos. Cuando un nodo pierde el arbitraje aplaza la transmisión de su trama para intentarlo de nuevo cuando finalice la trama actual. Conociendo los identificadores de todas las tramas que intentan ser transmitidas, se puede establecer de manera determinista el orden en el que son transmitidas. Así, una trama CAN con identificador más bajo (mayor número de bits dominantes en las primeras posiciones) tiene más prioridad que una trama con identificador más alto.

Tipos de Trama

Existen cuatro tipos de trama CAN:

- Trama de datos (data frame)
- Trama remota (remote frame)
- Trama de error (error frame)
- Trama de sobrecarga (overload frame)

Tipos de Datos

Una trama de datos CAN puede ser de uno de los dos siguientes formatos:

- Formato base: con identificador de 11 bits.
- Formato extendido: con identificador de 29 bits.

El estándar dice que un controlador CAN debe aceptar tramas en formato base, y puede o no aceptar tramas en formato extendido. Pero en cualquier caso debe tolerar tramas en formato extendido. Es decir, que si un controlador está configurado para que sólo acepte tramas en formato base no debe lanzar un error cuando reciba una trama en formato extendido, sino que simplemente no transmitirá el mensaje al procesador central.

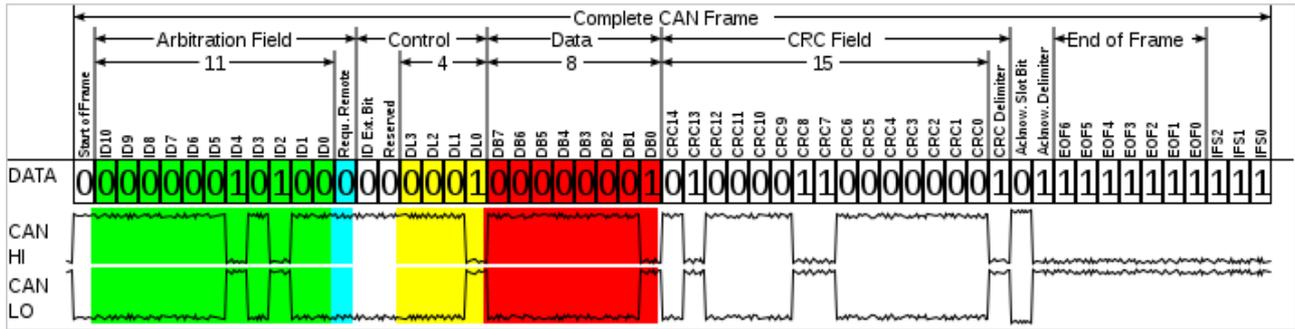


Figura 13: Trama CanBus.

Nombre del campo	Longitud (bits)	Finalidad
Inicio de trama	1	Demarca el comienzo de una transmisión.
Identificador - ID (verde)	11	Un identificador (único) que también representa la prioridad de la trama.
Petición de transmisión remota - RTR (cián)	1	Dominante (0) para tramas de datos y recesivo (1) para tramas de peticiones remotas.
Bit de extensión de identificador - IDE	1	Dominante (0) para el formato base (identificador de 11 bits).
Bit reservado (r0)	1	Bit reservado. Debe ser dominante (0), pero aceptado tanto dominante como recesivo.
Código de longitud de datos - DLC (amarillo)	4	Número de bytes de datos en el mensaje, entre 0 y 8. Si este campo es mayor que 8 el mensaje será de 8 bytes como máximo de cualquier modo.
Campo de datos (rojo)	0-64 (0-8 bytes)	Datos de la trama (la longitud del campo viene dada por el código de longitud de datos o DLC).
CRC	15	Verificación por redundancia cíclica. Código que verifica que los datos fueron transmitidos correctamente.
Delimitador CRC	1	Debe ser recesivo (1).
Hueco de acuse de recibo - ACK	1	El transmisor emite recesivo (1) y cualquier receptor emite dominante (0).
Delimitador ACK	1	Debe ser recesivo (1).
Fin de trama EOF	7	Debe ser recesivo (1).

Figura 14: Campos Trama CanBus.

Trama remota

Generalmente los datos se transmiten como trama de datos. Sin embargo, es posible que un nodo requiera unos datos desde otro nodo. En ese caso, el primero puede enviar una trama remota para pedir el envío de algún dato. El nodo que requiere la información envía entonces una trama con una petición de transmisión remota (RTR = 1; recesivo). Las tramas remotas o de petición de transmisión remota sólo se diferencian de las tramas de datos en que las tramas remotas no tienen campo de datos.

Trama error

La trama de error es una trama especial que viola las reglas de formato de las tramas CAN. Se transmite cuando un nodo detecta un mensaje erróneo, y provoca que los demás nodos también transmitan una trama de error. Un complejo mecanismo de contadores de error integrado en el controlador asegura que un nodo no bloquee el bus con continuas tramas de error.

Trama de sobrecarga

Es similar a la trama de error en cuanto a que viola el formato de las tramas CAN. Es transmitida por un nodo que se encuentra muy ocupado y el bus proporciona entonces un retardo extra entre tramas.

Separación entre tramas

Las tramas de datos y remotas están separadas por al menos tres bits recesivos (1). Después de eso, si se detecta un bit dominante (0), es considerado como el inicio de una nueva trama. Las tramas de error y de sobrecarga no respetan el espaciado entre tramas.

Modos de Lectura OBDII (PID)

PID, son códigos para comunicarse con el sistema de diagnóstico a bordo de un vehículo, estos están definidos por la SAE J1979 y son:

- 01: Muestra los parámetros disponibles
- 02: Muestra los datos almacenados por evento
- 03: Muestra los códigos de fallas de diagnóstico
- 04: Borra los datos almacenados incluyendo los códigos de fallas.
- 05: Resultado de la prueba de monitoreo de sensores de oxigeno
- 06: Resultado de la prueba de monitoreo de componentes/sistemas
- 07: Muestra código de fallas detectados en el último ciclo de manejo o actual
- 08: Operación de control de los componentes/sistemas de a bordo
- 09: Solicitud de información del vehículo
- 0A: Código de fallas permanentes (borrados)

La Sociedad de Ingenieros Automotrices (SAE) definió códigos PID para cada modo de operación bajo el estándar J1939. El fabricante del vehículo no está obligado a implementar todos los modos de operación o códigos, y tiene la libertad de añadir sus propios.

El técnico automotriz usa un dispositivo, escáner, para comunicarse con el sistema OBD-II, lo cual le permite obtener información o borrar los códigos de falla. Algunos escáneres tienen interfaces “amigables al humano”, con descripciones y gráficos. Otros dan acceso básico al sistema usando números binarios o hexadecimales. Este último tipo de comunicación es el que se describe a continuación.

Los bytes de respuesta se representan con las letras A, B, C, etc. A es el byte más significativo. Los bits de cada byte se representan del más significativo al menos con los números del 7 al 0.

A								B								C								D							
A7	A6	A5	A4	A3	A2	A1	A0	B7	B6	B5	B4	B3	B2	B1	B0	C7	C6	C5	C4	C3	C2	C1	C0	D7	D6	D5	D4	D3	D2	D1	D0

Figura 15: Formato Trama OBDII

Existen varios parámetros que pueden ser consultados, todos los pertenecientes al estándar, pero para el desarrollo de esta tesis solo me base en los siguientes, a su vez el único modo de operación que se usó es el 01 (Muestra los datos disponibles):

- Parámetro 04: Carga del motor (Torque).
- Parámetro 05: Temperatura del refrigerante.
- Parámetro 0C: RPM (Revoluciones por minuto).
- Parámetro 0D: Velocidad del vehículo.
- Parámetro 11: Posición del acelerador.

Si bien el integrado está preparado para trabajar con los cinco protocolos estandarizados del OBDII, para el desarrollo de esta tesis solo implementé dos que son el K- Line y el CANBUS, la justificación radica en que el CANBUS en un futuro será el estándar y por otra parte el K-Line porque el vehículo personal mío usa este protocolo para comunicación con la ECU.

1.1.4 Módulo Central:

- Realizado con PIC 18F46K80

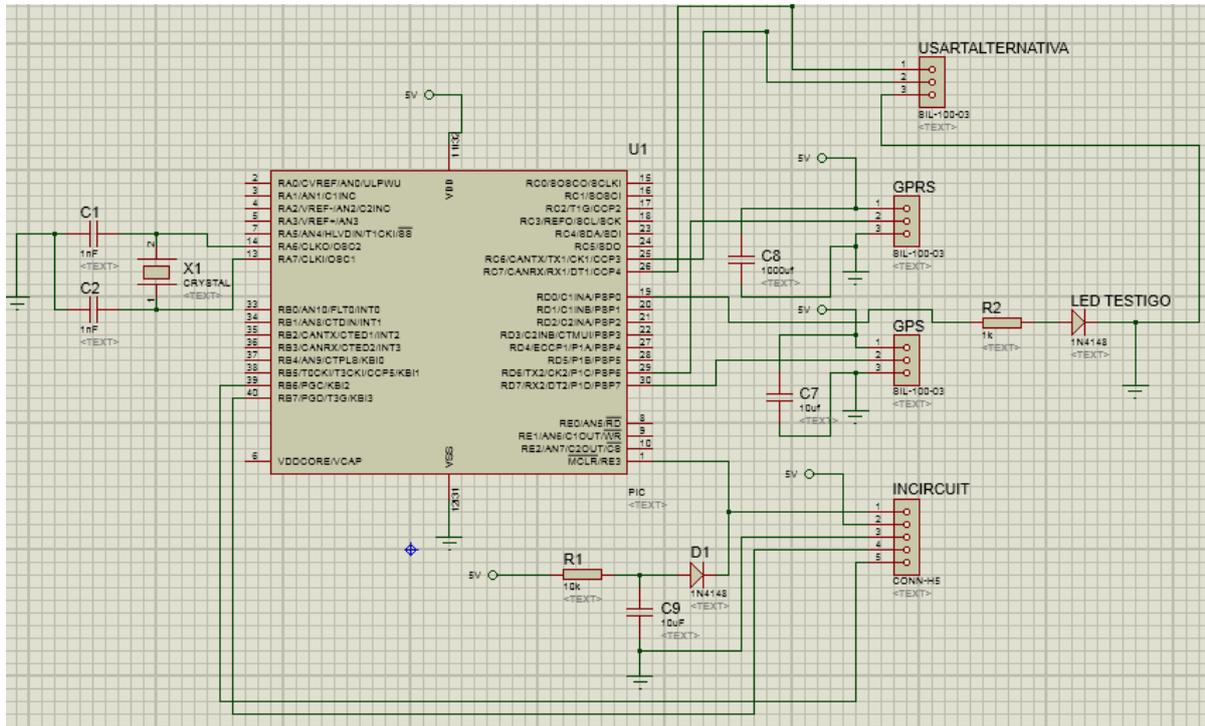


Figura 15: Esquema del microcontrolador.

Circuito implementado:

A continuación se detallan los circuitos finales implementados, estos se muestran por separada ya que las placas fueron realizadas por separada.

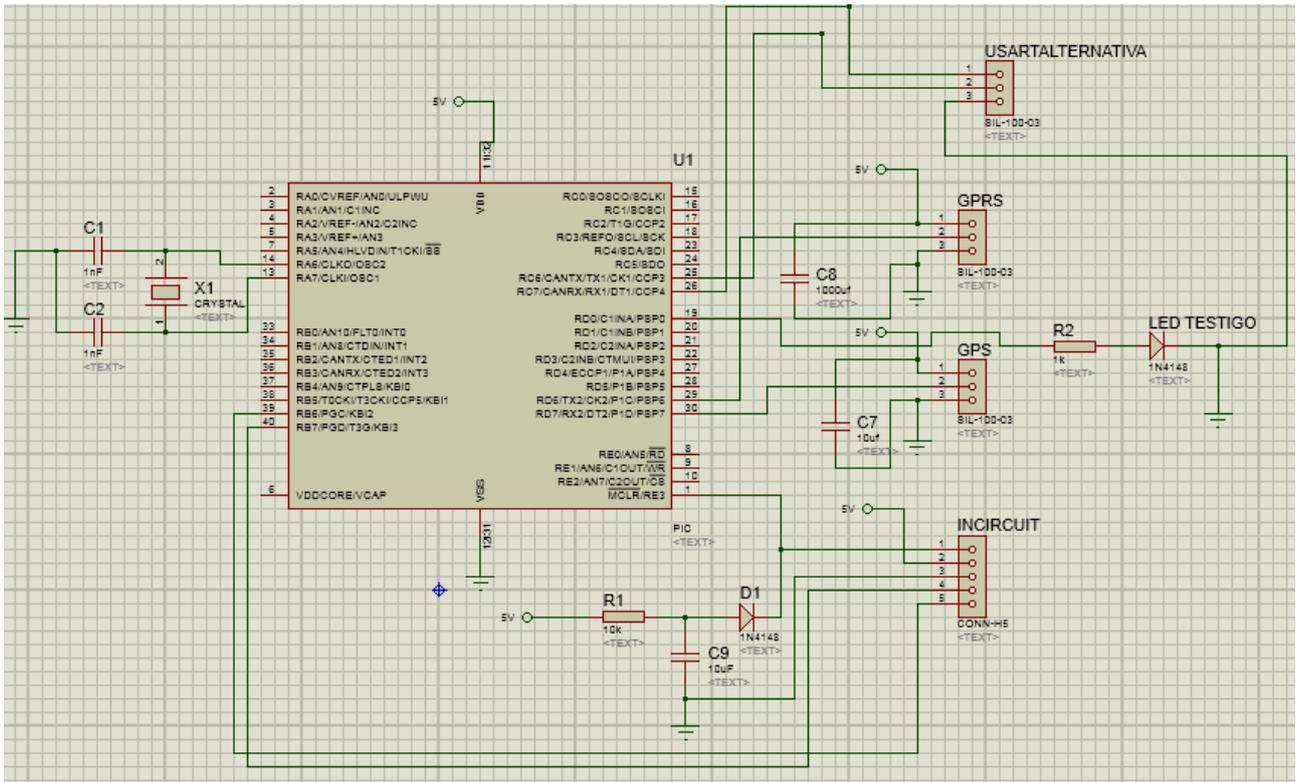


Figura 16: Placa Central.

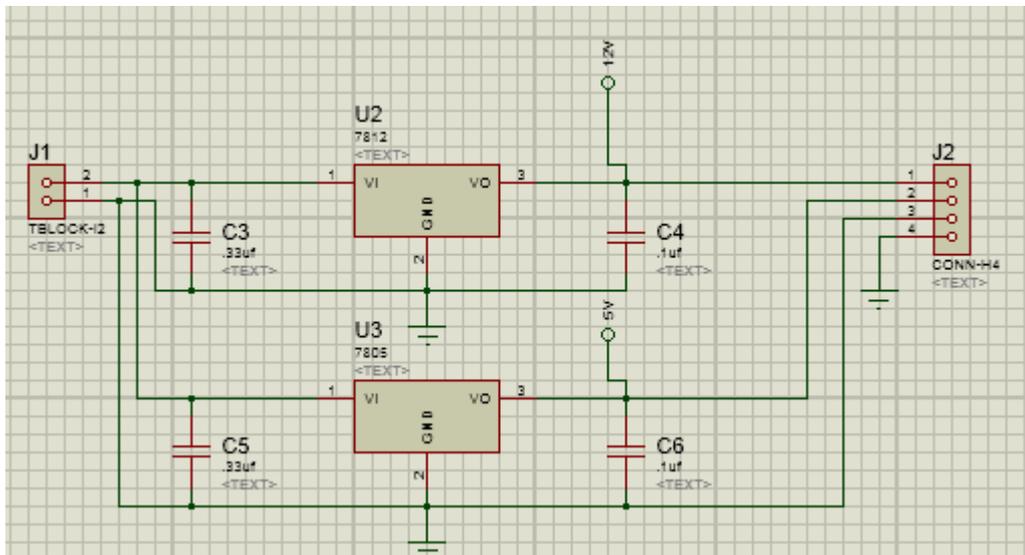


Figura 17: Placa de alimentación.

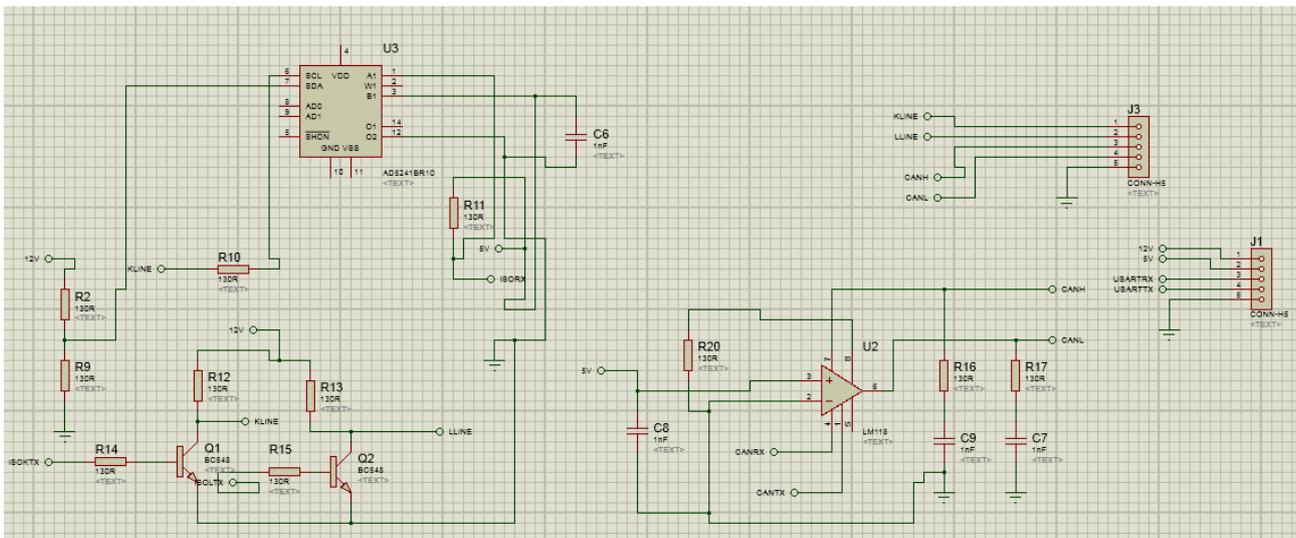
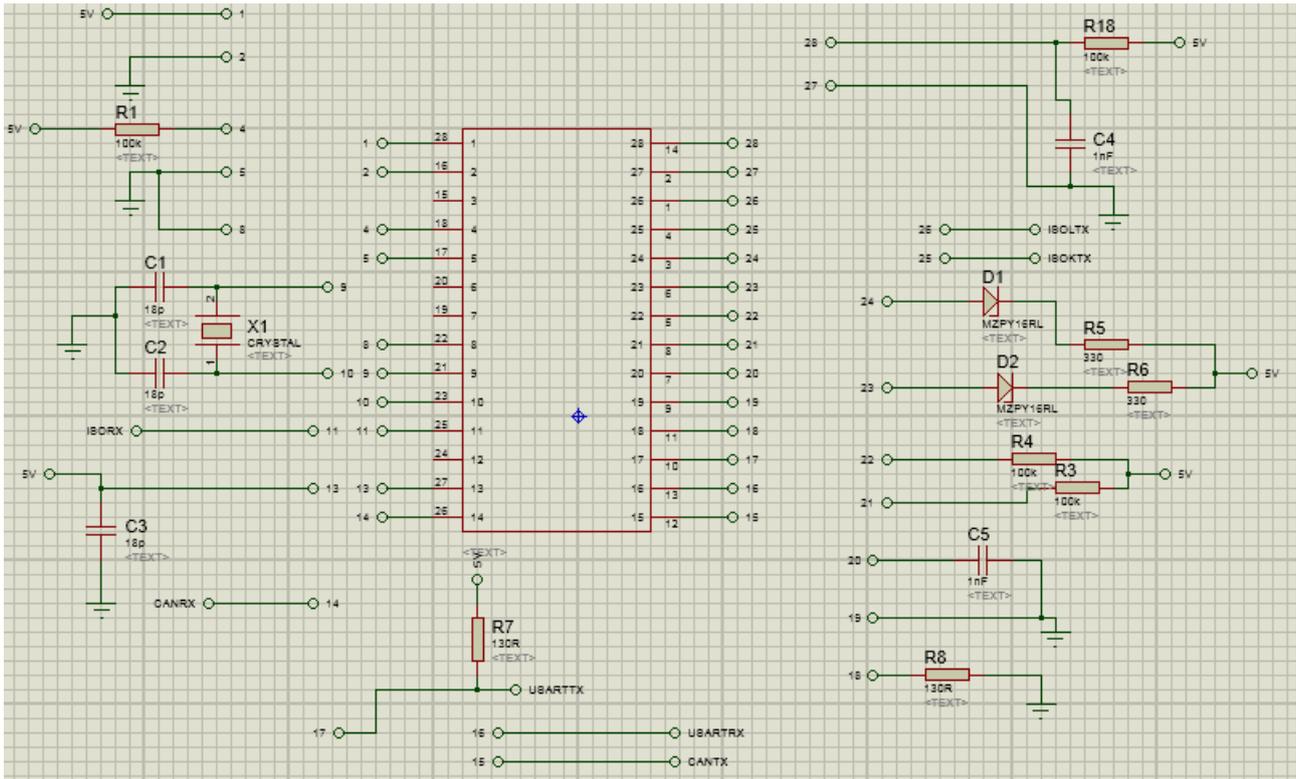


Figura 18: Placa Adquisidora (OBDII).

Problemas y soluciones implementadas

- Los GPS comerciales cuentan con muchos saltos de posición que deben ser filtrados por medio del firmware a implementar, filtré cambios bruscos de velocidad, cambios bruscos de rumbo, cambios de posición exagerados en poco tiempo.

- Cuando los vehículos están bajo techo el GPS calcula mal la posición debido a tener bloqueado la “vista” de los satélites, por lo que, por medio del firmware y los datos OBDII, cuando se detecta que el vehículo está detenido (Velocidad 0km/h) entonces se dejó de actualizar la posición.
- Como se trabaja con protocolo UDP y no se espera confirmación del paquete enviado al servidor, muchas tramas de datos se perdían por lo que implementé redundancia en la comunicación, reenviando 3 veces cada trama.
- Si bien el módulo GPS como GPRS no consumen mucho cuando están activos, si suelen tener picos de corriente cuando se posiciona o cuando se desea transmitir un dato, por lo que la etapa de alimentación calentaba un poco, entonces se optó por colocar un disipador para evitar inconvenientes y rediseños de la placa.
- El pickit 2 no soporta el microcontrolador usado para el desarrollo de mi tesis por lo que se debió generar archivos de configuración para que sea admitido y programado este tipo de dispositivos.

Análisis del funcionamiento

Placa Maestra:

Esta es la placa central del proyecto, es la encargada de realizar las peticiones necesarias para el relevamiento de datos de los distintos periféricos y combinarlos para enviarlos por medio del módulo GPRS al servidor.

La placa maestra cuenta con una interrupción por tiempo para indicar cuando se debe realizar una petición de variables a la ECU del auto, las peticiones son enviadas por el puerto serial hacia la placa de adquisidora de datos que a su vez transfiere estas solicitudes al puerto OBDII del vehículo.

Por el proceso inverso, del puerto OBDII salen las respuestas que son interpretadas por la placa adquisidora y enviadas a la placa Maestra que convierte los datos y los guarda en sus respectivas variables.

La recepción de datos desde la placa adquisidora se realiza mediante una interrupción por recepción de usart.

El otro puerto serial del microcontrolador está destinado a recibir las tramas enviadas por el GPS para procesarlas y guardar los datos correspondientes en variables manipulables, mientras que la línea de transmisión se utiliza para que el micro envíe las tramas formadas con los datos relevados hacia el módulo SIM800L que se encarga de enviarlas hacia el servidor. Previo a enviar la trama, realiza la configuración del modem para que cumpla el funcionamiento de transmisión GPRS.

Los datos del GPS son relevados mediante interrupciones por recepción de la usart, el módulo envía varias tramas (todas las correspondientes al protocolo NMEA) con distinto tipo de información, pero las necesarias para cubrir las necesidades de esta tesis son dos cuya cabecera son las RMC y GGA, esto me permite obtener las siguientes variables: UTC time, latitud, longitu, hdop, cantidad de satélites, altitud, azimuth, velocidad y calidad de la posición (2D o 3D).

Placa adquisidora:

Esta placa basa su funcionamiento en el integrado STN1110 que está diseñado para soportar (mediante una circuitería determinada por el fabricante) los protocolos estándares del OBDII. Para el desarrollo de este proyecto solo tuve en cuenta dos protocolos, el K-Line y el CanBus.

Desarrollo del software

- I. Este proyecto cuenta con una plataforma para administrar los datos relevados por el hardware, por una cuestión de portabilidad y compatibilidad con cualquier sistema operativo, se decidí realizar una web para que cumpla con los requerimientos.

La misma está desarrollada bajo PHP y cuenta con las siguientes partes:

- Mapa de google (google maps API) destinada a mostrar en tiempo real la posición del vehículo, la ventaja de utilizar esta api es la posibilidad de contar con todos los servicios de google incluyendo el Street view.
- Campo de datos en tiempo real, diseñado para mostrar los datos del vehículo, como lo son los datos GPS y variables tomadas desde la ECU.
- Campo para solicitud de históricos, aquí se puede solicitar a la base de datos un recorrido realizado por la unidad, este permite la generación de un archivo kml (compatible con google earth) donde muestro el recorrido solicitado.

- Campo sección de alarmas, aquí se van listando las últimas 5 alarmas generadas sobre la unidad, provenientes por excesos en las variables de la ECU, por ejemplo si se configura en el firmware un límite de 3500 RPM entonces cuando se detecte que se lee un valor mayor al predeterminado, entonces se visualiza la alarma en la web.

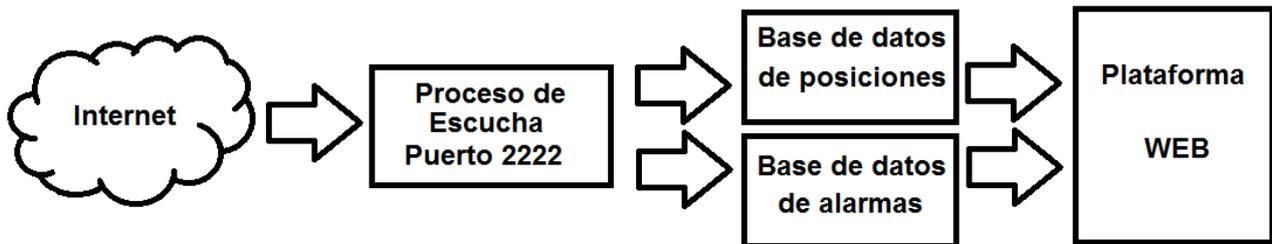


Figura 19: Esquema general del servidor.

Para poder tener una web montada necesité generar un servidor que la soporte, para este proyecto se utilicé Xampp que a su vez ya pone a disposición una base de datos SQL que es la encargada de guardar todos los datos que lleguen desde el equipo. La estructura diseñada está basada en tener 2 tablas, una para el posicionamiento en conjunto con los datos de la ECU, desde donde se levantan los datos para refrescar tanto la sección de mapa como la de variables en tiempo real y por otra parte se tiene una tabla encargada de guardar las alarmas generadas en el equipo, esta tabla solo alimenta la sección de alarmas.

Por último, para alimentar la web, hace falta un nexo entre los datos que llegan al servidor y la base de datos, a esto se le denomina negocio, consiste en un proceso encargado de escuchar el puerto al cual reportan los equipos, levanta las tramas correspondientes y realiza un parseo, luego de eso se inserta cada dato parseado en la posición de la tabla que le corresponde.

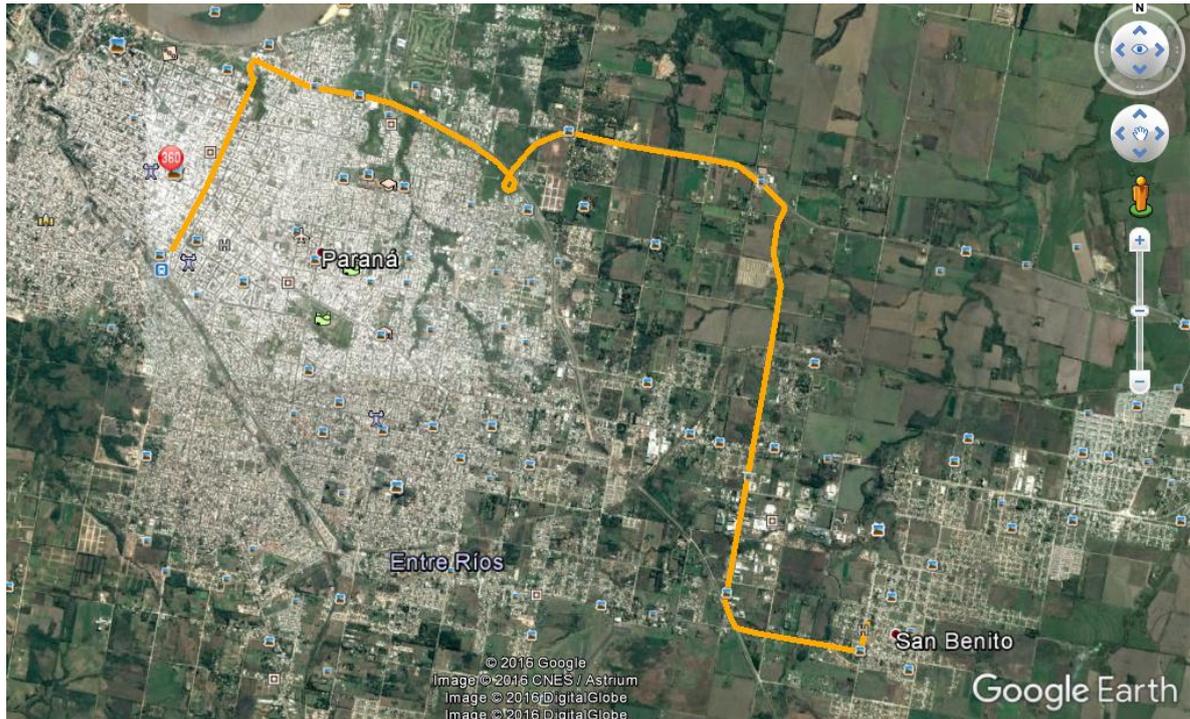


Figura 20: Recorrido generado sobre archivo KML.

1.2 Diseño Completo

Finalmente cuento con un equipo de fácil instalación y operación, el dispositivo solo debe ser conectado al puerto OBDII del vehículo y este comenzará a operar cada vez que el auto se ponga en marcha, esto tiene dos ventajas, primero que no consume batería del auto cuando este no está en marcha por lo que implica que no tendremos problema de desgaste o agotamiento de la misma y por otra parte evita saltos de posicionamiento cuando el automotor se encuentra detenido. El dispositivo puede ser oculto dentro del tablero del auto para que sea imperceptible para el conductor.

Por otra parte el producto queda resguardado mediante una caja estanco lo que lo protege de cualquier tipo de partícula o líquido que pueda afectar a la electrónica del mismo.

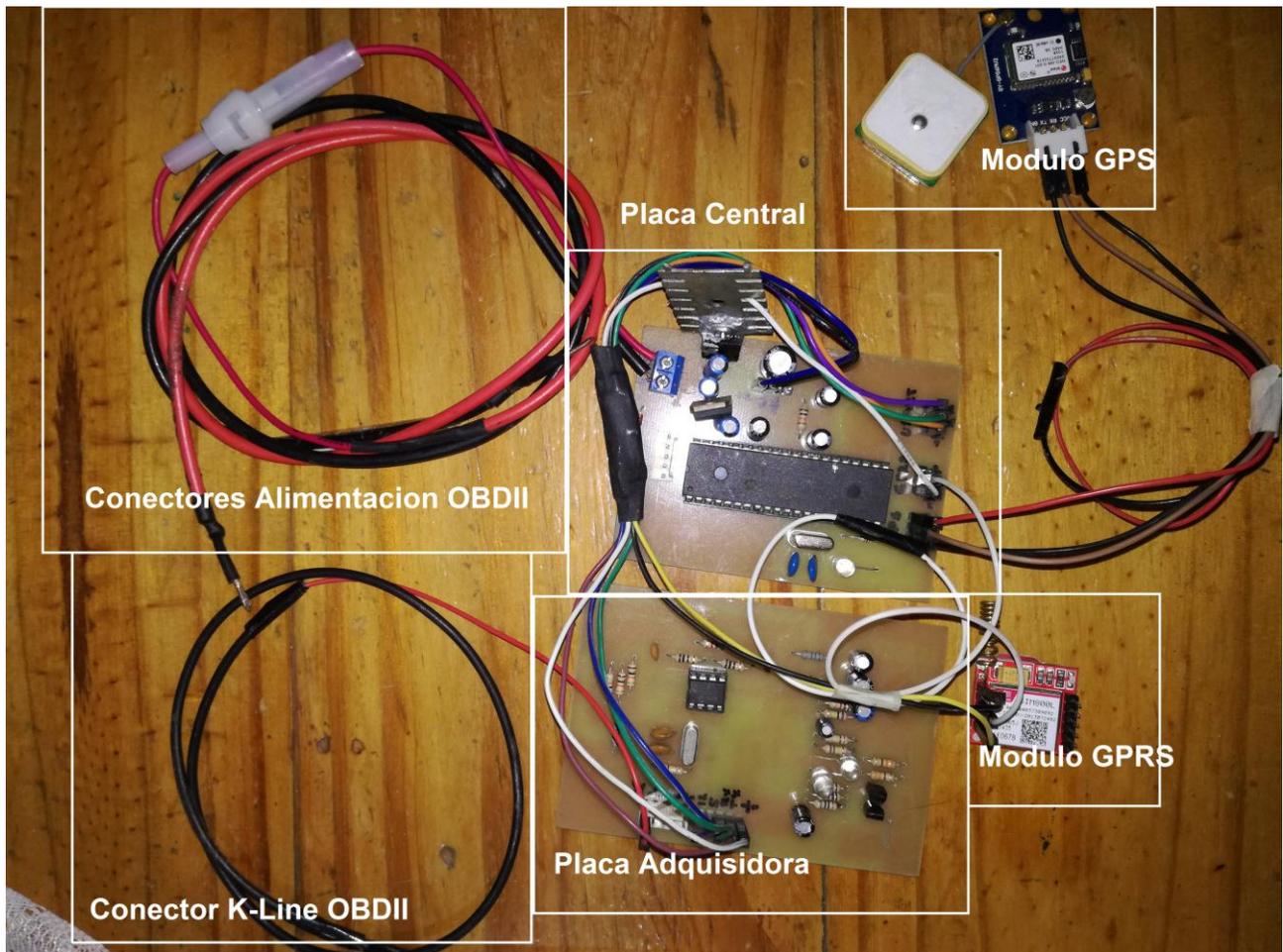


Figura 21: Producto final.

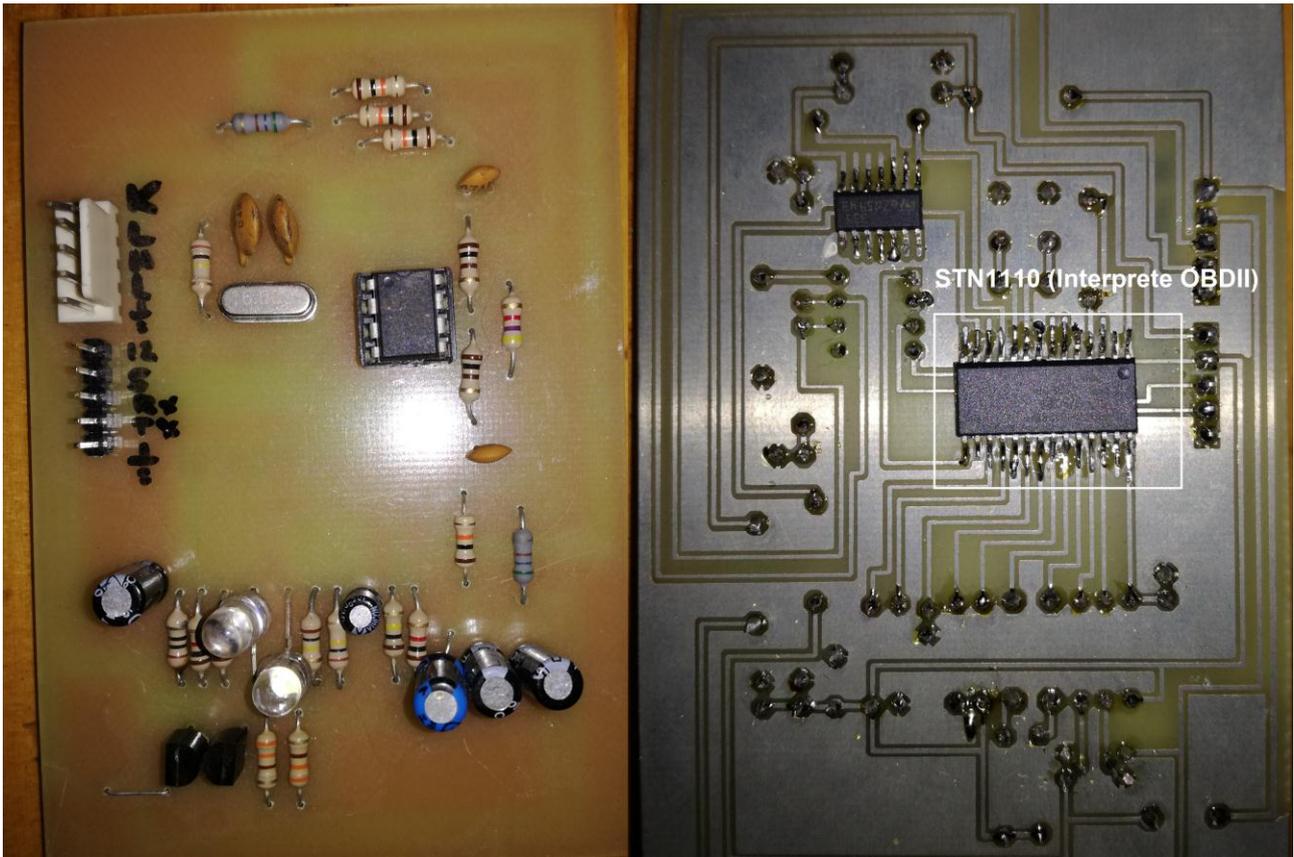


Figura 22: Placa adquisidora (Intérprete OBDII).

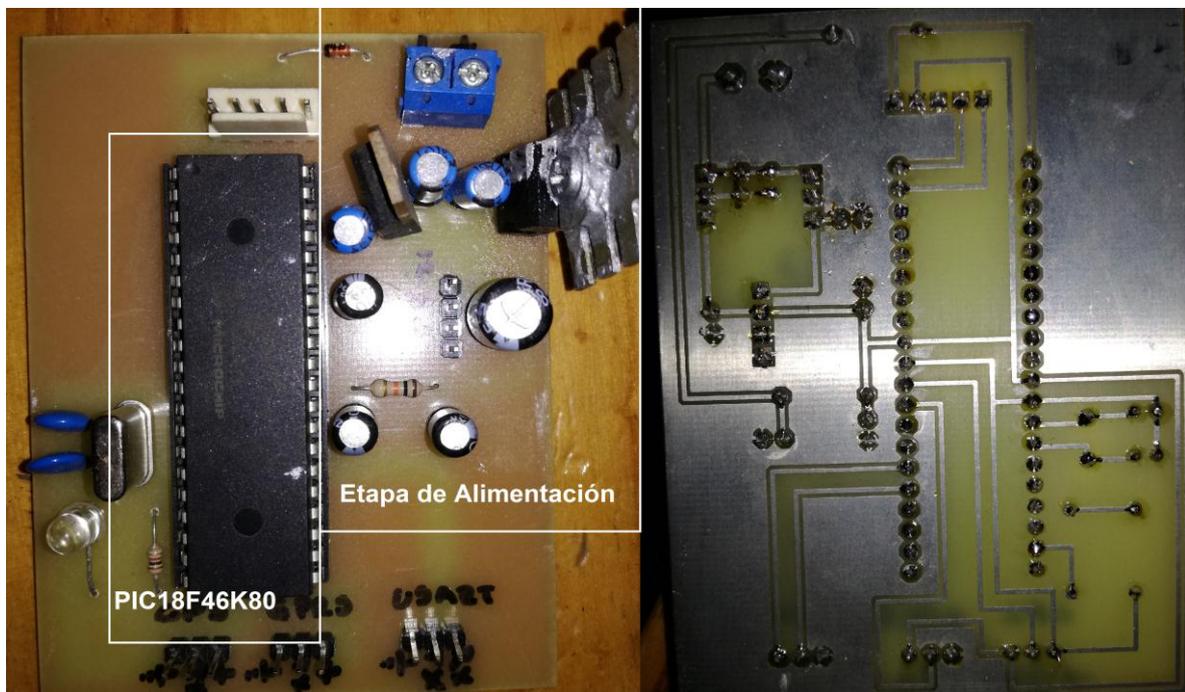


Figura 24: Placa Central

El sistema implementado brinda al usuario una amplia gama de datos que no son normalmente ofrecidos por las empresas dedicadas a la logística, el usuario final puede contar en tiempo real con los datos de posicionamiento del vehículo que le permitan saber dónde se encuentra su flota en cada momento o bien por medio de la herramienta de solicitud de históricos se puede acceder al posicionamiento antiguo de la unidad.

A su vez, el cliente puede controlar las conductas de manejo de sus choferes o de quien manipule su vehículo, para de esta manera pulir las formas de conducir para obtener el mejor rendimiento del automotor y evitar así gastos innecesarios de reparaciones o servicios que deban realizarse, como así también se obtienen mejores performances en el consumo de combustible.

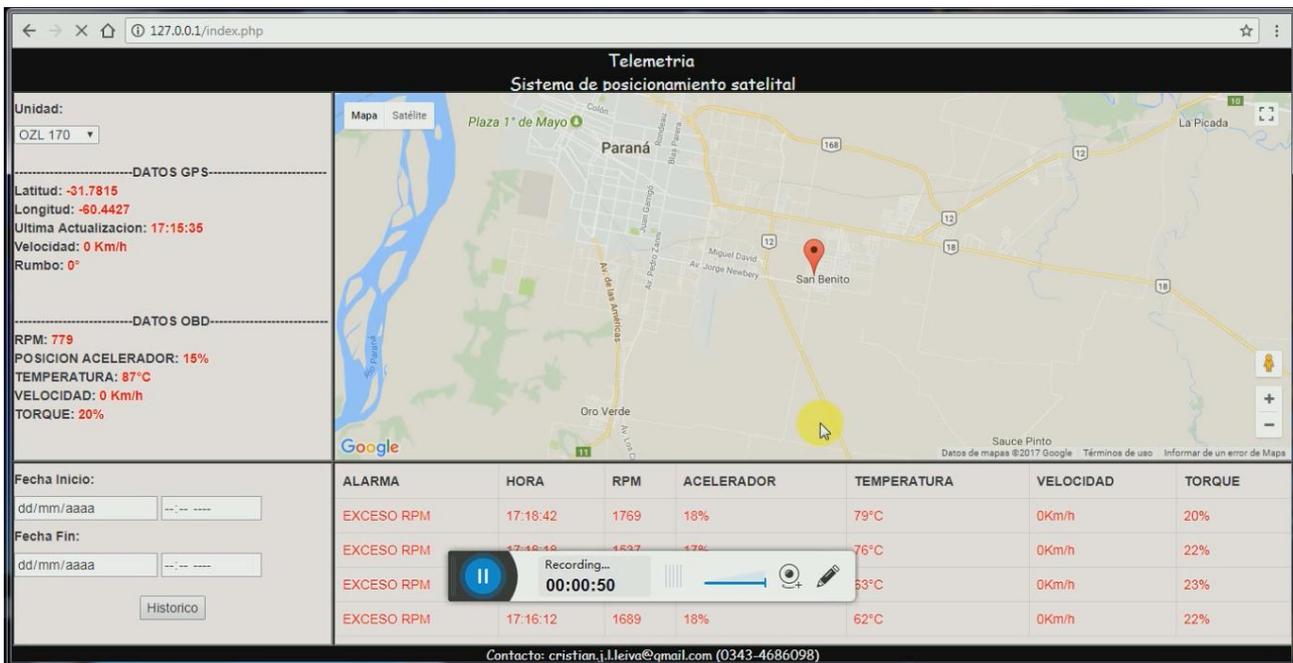


Figura 23: Web Fina

Capítulo 3: Resultados

El sistema desarrollado cuenta con varias prestaciones que hoy en día existen en el mercado tanto nacional como internacional, desde el punto de vista del servicio de logística basta con buscar y se encontrará uno con varias empresas que se dedican a prestar este tipo de servicios, sin embargo estas lo brindan en su mayoría con equipos importados, se presenta aquí entonces la primer ventaja, dependiendo de las limitantes que presente la nación para la importación puede ser una gran opción utilizar un equipo de diseño y producción nacional.

Otra ventaja que brinda esta solución es la posibilidad de dar seguimiento web, no todas las empresas de rastreo ponen a disposición un servicio web, sino que ofrecen la custodia y seguimiento de las unidades, el objetivo de mi tesis es que cada usuario pueda encargarse de su propia flota solo solicitando revisiones técnicas o mantenimientos.

Como desventaja podemos aclarar que falta aún mucho diseño y mejora en cuanto a hardware y firmware, obviamente esto en comparación con equipos que tienen más de 10 años de ingeniería invertidos en ellos, sin embargo la lógica o la forma de plantear la solución no es tan desventajosa comparada con estos dispositivos comerciales.

Una nueva ventaja son los costos, importar un equipo de este tipo cuesta 1000 dolares siendo que esto es solo el hardware, se debe contar con una empresa que te ofrezca el servicio de la plataforma para poder rastrearlo, por lo que en este capítulo, si bien no se profundiza en el costo del dispositivo, si puede darse la idea de que será mucho más económico desde el punto de vista costo-servicio que se presta.

Capítulo 4: Análisis de Costos

No realicé para este proyecto un seguimiento exhaustivo de las horas implementadas para el desarrollo e implementación, sin embargo si se puede dimensionar el trabajo realizado por el desarrollo en sí. Vale aclarar nuevamente, como se mencioné en el capítulo anterior, que el hardware realizado en comparación con equipos comerciales está aún bastante alejado en cuanto a robustez, pero esto se debe al tiempo de dedicación que aún necesita este producto para llegar a estar insertado en el mercado.

Sin embargo, dado que estoy en conciencia de que el equipo aun no puede ser comparado con los equipos importados, si cumple este con varias funciones que sirven para brindar un servicio de rastreo con combinación de control de variables de la ECU para revisar las conductas de manejo de quien utiliza los vehículos, por lo que este desarrollo, en cuanto a hardware aun es un prototipo que debe ser pulido y aun mas desarrolla, por lo que el costo que debe asumirse en el hardware claramente no puede superar el costo comercial de un producto importado.

Se presenta a continuación un listado de costo de material y producción del hardware:

Insumo	Costo
Modulo GPS	\$450
Modulo GPRS	\$350
Integrado STN1110	\$90
PIC 18F46K80	\$75
Componentes electrónicos varios	\$250
Fabricación PCB	\$100
Ensamble	\$150
Total	\$1465

Tabla 3: Costos.

El desarrollo de hardware consumió un tiempo de 150hs, en cuanto al software de seguimiento, el tiempo implementado ronda en las 100 hs, con un costo promedio de hora laboral de \$175, se calcula que entre desarrollo y costo de material, este proyecto ronda (solo el prototipo) en \$50.000, lo que equivale a unos 2.800 dolares.

Como mencione anteriormente, no se pretende con estos números poner un costo al producto ya que no hay margen de comparación entre un prototipo y un equipo importado.

El costo indicado anteriormente hace referencia al costo de desarrollo de este prototipo no a un costo de comercialización del mismo.

Capítulo 5: Discusión y Conclusión.

El fin de este proyecto fue lograr un prototipo de equipo capaz de posicionar y relevar parámetros de la ECU para ser brindados a un cliente que pueda interpretarlos y usarlos a su gusto, logré un dispositivo básico pero suficiente para demostrar que se puede cumplir el objetivo que luego de varias mejoras puede competir tranquilamente contra los equipos que por importación pueden conseguirse, se listan a continuación las mejoras propuestas para continuar avanzando en este proyecto:

- Implementación de un protocolo mas robusto de comunicación que cuente con detección de caídas de paquetes, retransmita, almacene y compruebe tramas erróneas.
- Implementación de una interfaz de usuario para poder configurar los equipos de manera que cada cliente pueda configurar sus alarmas y tasas de reportes a gusto.
- Implementación de un sistema de actualización de firmware remota.
- Implementación de filtro de tramas dañadas en el módulo que inserta valores en la Base de datos del servidor.
- Implementación de un sistema de alimentación de backup para que el equipo pueda seguir reportando cuando se quita el contacto.
- Implementación de un sistema de filtrado tanto el firmware como software para casos de saltos de posición.

Con las mejoras antes descriptas puedo tener un equipo que sea más competitivo con los de origen internacional.

Capítulo 6: Literatura Citada.

PID OBDII. https://es.wikipedia.org/wiki/OBD-II_PID

PHP. <https://www.php.net>

Protocolo NMEA. <http://www.gpsinformation.org/dale/nmea.htm>

SIM800L Datasheet. img.filipeflop.com/files/download/Datasheet_SIM800L.pdf

GPS Ublox NEO 6M. [https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_\(GPS.G6-HW-09005\).pdf](https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_(GPS.G6-HW-09005).pdf)

PIC 18F46K80 Datasheet. www.microchip.com/PIC18F46K80

STN1110 Datasheet. https://www.scantool.net/scantool/.../102/stn1110_vs_elm327.pdf

MCP2551 Datasheet. www.datasheetspdf.com/datasheet/MCP2551.html

Universidad Tecnológica Nacional
Facultad Regional Paraná

Se certifica que , DNI: ha realizado la
dirección del Proyecto Final:

.....
.....
.....

De los alumnos:

-
-
-

Realizada durante el ciclo lectivo:, obteniendo el grupo un calificación final de:

.....

A fin de ser emitida la correspondiente certificación por el departamento de electrónica, se extiende
la siguiente constancia.

Pañoni Sergio

Ramos Hector

Maggiolini Lucas