

**DISEÑO Y DESARROLLO DE UN DISPOSITIVO DE HARDWARE PARA EL
MONITOREO Y DIAGNÓSTICO DE AUTOMÓVILES BASADOS EN CAN BUS**



**JAIRO ADRIAN CAICEDO BRAVO
JOSÉ CAMILO ERASO GUERRERO**

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE ELECTRÓNICA
SAN JUAN DE PASTO**

2017

**DISEÑO Y DESARROLLO DE UN DISPOSITIVO DE HARDWARE PARA EL
MONITOREO Y DIAGNÓSTICO DE AUTOMÓVILES BASADOS EN CAN BUS**



**JAIRO ADRIAN CAICEDO BRAVO
JOSÉ CAMILO ERASO GUERRERO**

Trabajo de grado para optar por el título de Ingeniería Electrónica

**ASESOR
Mg. JUAN CARLOS CASTILLO ERASO**

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE ELECTRÓNICA
SAN JUAN DE PASTO**

2017

NOTA DE RESPONSABILIDAD

“La Universidad de Nariño no se hace responsable por las opiniones o resultados obtenidos en el presente trabajo y para su publicación priman las normas sobre el derecho de autor.”

Acuerdo 1. Artículo 324. Octubre 11 de 1966, emanado del honorable Consejo Directivo de la Universidad de Nariño.

NOTA DE ACEPTACIÓN

Firma del Presidente del Jurado

Firma Jurado 1

Firma Jurado 2

San Juan De Pasto, 24 de Febrero de 2017

DEDICATORIA

Esta tesis y este gran logro en mi vida se lo dedico a Dios Todopoderoso, mi alabanza de gratitud por bendecirme con sabiduría y entendimiento.

Dedicado a todos mis seres queridos por darme su mayor apoyo incondicional, consejos, ánimo y compañía, que han logrado forjar mi futuro. Algunos están aquí conmigo, otros en mi recuerdo y mi corazón.

A ti María, mi madre, no hay palabras suficientes, todo mi agradecimiento, por estar ahí en todo momento, por tus consejos, tus valores y motivación para ser una persona de bien, por el ejemplo, la responsabilidad, por el valor y la tenacidad para salir adelante. Por ser el pilar fundamental en todo lo que soy, por eso que lo es todo, pero más que nada por tu amor.

Para ti Mary, por ser parte de mi hoy y mi futuro, por nuestro hijo Jacob, motivación para seguir triunfando juntos.

José E

A la memoria de mi padre Jairo Caicedo quien ha sido mi fuente de inspiración y el motor que me impulsa a ser una gran persona.

A mi madre, Lilia Bravo que me ha regalado su amor, confianza y apoyo para que ahora culmine esta etapa de mi vida.

A mis queridas hermanas, Ayda Lilia y Nidia Yadira que siempre están a mi lado brindándome su ayuda y dándome ánimos para seguir construyendo y haciendo realidad mis sueños

Finalmente a toda mi familia que de una u otra manera me han motivado para que día tras día insista y pueda alcanzar esta meta.

Jairo C

AGRADECIMIENTOS

Agradecemos a la Universidad de Nariño por darnos la oportunidad de haber sido estudiantes Liceístas y hoy profesionales; aprovechando la idoneidad, experiencia y competencia de tantos maestros que impulsaron nuestra formación.

A usted, nuestro director de tesis Mg. Juan Carlos Castillo Eraso por su invaluable apoyo y valiosa asesoría en plasmar nuestras ideas en este proyecto de tesis y culminarlo satisfactoriamente.

A todos nuestros compañeros por lo vivido y compartido, por esos increíbles y buenos ratos, por eso tan valioso que es nuestra amistad.

¡Gracias a ustedes!

CONTENIDO

INTRODUCCIÓN.....	15
1. DESCRIPCION DEL PROBLEMA	16
1.1 PLANTEAMIENTO DEL PROBLEMA	16
1.2 OBJETIVO GENERAL.....	16
1.3 OBJETIVOS ESPECÍFICOS	16
1.4 JUSTIFICACIÓN.....	17
2. MARCO TEORICO	18
2.1 SISTEMA DE POSICIONAMIENTO GLOBAL (GPS)	18
2.2 MICROCONTROLADOR	19
2.2.1 Plataforma ARDUINO	19
2.3 SISTEMAS DE COMUNICACIÓN	19
2.3.1 Tecnología GSM	20
2.3.1.1 Arquitectura de una Red GSM	20
2.3.2 Tecnología GPRS.....	22
2.3.2.2 Arquitectura de una Red GPRS	22
2.4 COMANDO HAYES O COMANDOS AT	23
2.5 PROTOCOLO CAN BUS.....	25
2.5.1 Características Básicas Del CAN BUS.....	27
2.5.1.1 Normas de Estandarización	28
2.5.1.2 Estructura Definida	28
2.5.1.3 Número de Nodos.....	28
2.5.1.4 Optimización del Ancho de Banda	28
2.5.1.5 Detección y Desactivación de Nodos Defectuosos	28
2.5.1.6 Relación Velocidad/Distancia.....	29
2.5.1.7 Recepción por Multidifusión (Multicast)	29
2.5.1.8 Jerarquía Multimaestro	30
2.5.2 Estructura de Capas del Protocolo CAN Bus	30
2.5.3 Tramas Protocolo CAN Bus	31
2.5.3.1 Tipos de Tramas	31

2.5.3.2 Validación de Tramas	33
2.5.4 Proceso para la Transmisión de Datos	33
2.5.4.1 Suministro de Datos	33
2.5.4.2 Transmisión de Datos	34
2.5.4.3 Recepción de Datos	34
2.6 SISTEMA DE DIAGNÓSTICO OBD-II	35
2.6.1 Modos de Prueba del OBD-II	38
2.6.2 Códigos de Diagnóstico Fallas DTC	38
2.6.3 Conector de Diagnóstico Estándar	40
2.6.4 Cable Conector Entre Puerto OBD-II y Tarjeta ARGUS AUTOMOTIVE	41
3. DISEÑO DEL DISPOSITIVO ARGUS AUTOMOTIVE LM8095	42
3.1 MICROCONTROLADOR ATMEGA328P-PU	44
3.1.1 Esquema de Conexión del Microcontrolador ATmega328P-PU	45
3.2 MÓDULO GPS QUECTEL L80	48
3.2.1 Hardware Quectel L80	50
3.2.2 Protocolo de Comunicación para Quectel L80	50
3.2.4 Esquema de Conexión del Receptor GPS Quectel L80	53
3.3 MÓDULO GSM/GPRS QUECTEL M95	54
3.3.1 Hardware Quectel M95	56
3.3.2 Protocolo de Comunicación para Quectel M95	58
3.3.3 Esquema de Conexión del Módulo GSM/GPRS Quectel M95	58
3.3.3.1 Etapa de Potencia para el Módulo Quectel M95	58
3.3.3.2 Indicadores de STATUS y NETLIGHT	59
3.3.3.3 Interfaz SIM Card del Módulo Quectel M95	60
3.3.3.4 Interfaz de Comunicación UART para el Quectel M95	60
3.3.3.4.1 Etapa de Acople de Voltaje por Medio de Transistores 2N7000	61
3.4 DESARROLLO DEL SISTEMA CAN BUS	63
3.4.1 Controlador CAN MCP2515	64
3.4.2 Transceptor MCP2551	67
3.4.3 Esquema de Conexión CAN Bus	69
3.4.3.1 Conexión del Controlador MCP2515	69
3.4.3.2 Conexión del Transceptor MCP2551	70

4. ESQUEMA Y FABRICACIÓN DEL DISPOSITIVO FINAL.....	74
4.1 MODOS DE FUNCIONAMIENTO DE LA TARJETA ARGUS AUTOMOTIVE	83
5. FIRMWARE ARGUS AUTOMOTIVE	84
5.1 FIRMWARE PARA OBTENER POSICIÓN GPS	85
5.2 FIRMWARE PARA MONITOREO Y DIAGNOSTICO	86
5.2.1 Firmware Monitoreo de Variables.....	86
5.2.2 Firmware de Diagnóstico	90
5.3 FIRMWARE PARA EL ENVIO DE DATOS GPRS.....	90
6. RESULTADOS	94
CONCLUSIONES.....	99
RECOMENDACIONES	100
REFERENCIAS	101
ANEXOS	105

LISTA DE TABLAS

TABLA 1. SINTAXIS EXTENDIDA DE COMANDOS AT	24
TABLA 2. RELACIÓN VELOCIDAD/DISTANCIA PARA EL PROTOCOLO CAN BUS	29
TABLA 3. ARQUITECTURA DE LAS CAPAS DEL PROTOCOLO CAN BUS	30
TABLA 4. DESCRIPCIÓN DE PINES DEL CONECTOR OBD-II	40
TABLA 5. CARACTERÍSTICAS DEL MICROCONTROLADOR ATMEGA328P-PU	44
TABLA 6. CARACTERÍSTICAS DEL MÓDULO QUECTEL L80	49
TABLA 7. DESCRIPCIÓN DE UNA TRAMA GPGGA	51
TABLA 8. DESCRIPCIÓN DE UNA TRAMA GPRMC	52
TABLA 9. PRINCIPALES CARACTERÍSTICAS DEL MÓDULO QUECTEL M95	55
TABLA 10. ESTADOS DE RED DEL MÓDULO QUECTEL M95	57
TABLA 11. PARÁMETROS DEL CONTROLADOR MCP2515	65
TABLA 12. HARDWARE MCP2551	68
TABLA 13. VALOR APROXIMADO TARJETA ARGUS AUTOMOTIVE	82
TABLA 14. VARIABLES A MONITOREAR DEL AUTOMÓVIL CON SU RESPECTIVO ID	87
TABLA 15. PARÁMETROS DE LAS VARIABLES DE MONITOREO	88

LISTA DE FIGURAS

FIGURA 1. CÁLCULO DE LA POSICIÓN DE UN RECEPTOR.....	18
FIGURA 2. ARQUITECTURA DE UNA RED GSM.....	21
FIGURA 3. ARQUITECTURA DE UNA RED GPRS.....	22
FIGURA 4. CONEXIÓN UNIDADES DE CONTROL SIN CAN BUS.....	25
FIGURA 5. CONEXIÓN ENTRE UNIDADES DE CONTROL CON CAN BUS.....	26
FIGURA 6. ESQUEMA DE COMPARACIÓN DEL CAN BUS CON EL MEDIO FÍSICO.....	27
FIGURA 7. FORMATO DE TRAMA.....	32
FIGURA 8. PROCESO DE TRANSMISIÓN DE DATOS EN EL AUTOMÓVIL.....	34
FIGURA 9. UBICACIÓN INDICADOR CHECK ENGINE EN AUTOMÓVILES.....	36
FIGURA 10. UBICACIÓN DEL CONECTOR OBD-II EN EL AUTOMÓVIL.....	37
FIGURA 11. FORMATO DE CÓDIGOS DE FALLA.....	39
FIGURA 12. CABLE CONECTOR CON TERMINALES OBD-II Y RS232.....	41
FIGURA 13. DIAGRAMA DE BLOQUES DE ARGUS AUTOMOTIVE.....	43
FIGURA 14. ESQUEMA DE CONEXIÓN DEL ATMEGA328P-PU.....	46
FIGURA 15. PRIMERA FABRICACIÓN DEL CIRCUITO PARA EL ATMEGA328P-PU.....	47
FIGURA 16. DIMENSIONES DEL MÓDULO QUECTEL L80.....	48
FIGURA 17. PINES DEL RECEPTOR GPS QUECTEL L80.....	50
FIGURA 18. ESQUEMA DE CONEXIÓN DEL RECEPTOR GPS QUECTEL L80.....	53
FIGURA 19. DIMENSIONES DEL MÓDULO QUECTEL M95.....	54
FIGURA 20. PINES DEL MÓDULO GPRS QUECTEL M95.....	56
FIGURA 21. ETAPA DE POTENCIA PARA EL MÓDULO QUECTEL M95.....	58
FIGURA 22. ESQUEMA DE INDICADORES DE LOS PINES STATUS Y NETLIGHT.....	59
FIGURA 23. ESQUEMA PARA LA INTERFAZ SIM CARD.....	60
FIGURA 24. ESQUEMA PARA EL ACOUPLE DE VOLTAJES.....	61
FIGURA 25. ACOUPLE ENTRE RX DEL QUECTEL M95 Y EL TX DEL MICROCONTROLADOR...62	62
FIGURA 26. ACOUPLE ENTRE TX DEL QUECTEL M95 Y EL RX DEL MICROCONTROLADOR...62	62
FIGURA 27. CIRCUITO DEL MÓDULO QUECTEL M95 EN PROTOBOARD.....	63
FIGURA 28. DIAGRAMA DE PINES DEL CONTROLADOR CAN MCP2515.....	64
FIGURA 29. DIAGRAMA DE PINES DE TRANSECTOR MCP2551.....	67
FIGURA 30. INTERACCIÓN ENTRE RED CAN BUS DEL AUTOMÓVIL Y MÓDULO CAN BUS..71	71
FIGURA 31. ESQUEMA DE CONEXIÓN CAN BUS.....	72
FIGURA 32. PRIMERA FABRICACIÓN DEL SISTEMA CAN BUS.....	74
FIGURA 33. ESQUEMA GENERAL ARGUS AUTOMOTIVE.....	75
FIGURA 34. CONECTOR FT232RL.....	76
FIGURA 35. MONTAJE EN PROTOBOARD DE ARGUS AUTOMOTIVE.....	76
FIGURA 36. ESQUEMA CIRCUITO IMPRESO EN PLACA PCB.....	77
FIGURA 37. PLACA PCB EN ÁCIDO FÉRRICO.....	77
FIGURA 38. PISTAS EN COBRE DEL ESQUEMA ARGUS AUTOMOTIVE.....	78
FIGURA 39. PRIMER PROTOTIPO TARJETA ARGUS AUTOMOTIVE.....	79
FIGURA 40. RUTAS DE ARGUS AUTOMOTIVE EN EAGLE.....	80
FIGURA 41. COTIZACIÓN PARA EL ENSAMBLE DE ARGUS AUTOMOTIVE.....	81
FIGURA 42. DISPOSITIVO FINAL.....	81
FIGURA 43. UBICACIÓN DE LOS JUMPERS EN EL DISPOSITIVO.....	83
FIGURA 44. MODELO DEL SISTEMA DEL FUNCIONAMIENTO DE ARGUS AUTOMOTIVE.....	84
FIGURA 45. DIAGRAMA DE FLUJO DEL FIRMWARE PARA EL POSICIONAMIENTO DEL GPS.....	86

FIGURA 46. FORMATO DE LA TRAMA RESPONDIDA POR LA RED CAN BUS	87
FIGURA 47. DIAGRAMA DE FLUJO PARA EL MONITOREO DEL AUTOMÓVIL.....	89
FIGURA 48. DIAGRAMA DE FLUJO PARA EL DIAGNÓSTICO DEL AUTOMÓVIL	90
FIGURA 49. DIAGRAMA DE FLUJO PARA LA TRANSMISIÓN DE DATOS	91
FIGURA 50. PÁGINA WEB DEL DISPOSITIVO	94
FIGURA 51. RECEPCIÓN DE LOS DATOS DEL AUTOMÓVIL EN EL SERVIDOR	95
FIGURA 52. UBICACIÓN DEL AUTOMÓVIL EN EL MAPA.....	96
FIGURA 53. TABLERO DE INSTRUMENTOS EN PÁGINA WEB.....	97
FIGURA 54. BASE DE DATOS.....	97
FIGURA 55. ESTRUCTURA DE UNA TRAMA EN LA BASE DE DATOS.....	98
FIGURA 56. PROCESO DE ARBITRAJE PARA LA TRANSMISIÓN DE DATOS.....	110
FIGURA 57. MEDIO FÍSICO DEL CAN BUS.....	112
FIGURA 58. NIVELES DE TENSIÓN, (ESTÁNDAR ISO11898)	113
FIGURA 59. CONTROLADORES DE LOS COMPONENTES DEL CAN BUS	114
FIGURA 60. TRANSCPTORES DE RED CAN BUS.....	115
FIGURA 61. MENSAJE DE TRANSMISIÓN REMOTA.....	116
FIGURA 62. TRAMA DE ERROR.....	117

LISTA DE ANEXOS

ANEXO 1.....	102
ANEXO 2.....	105
ANEXO 3.....	113
ANEXO 4.....	116

RESUMEN

Es evidente el crecimiento de las ciudades colombianas, lo que obliga a las personas a transitar mayores distancias para movilizarse, esto explica el aumento en ventas de automóviles en el país. Es de resaltar, que también han aumentado el número de accidentes de tránsito en Colombia, gran parte de estos incidentes de vehículos, se debe a excesos de velocidad por parte de los conductores o a fallas en el funcionamiento de los autos. Se plantea en la investigación el diseño y construcción de un dispositivo de hardware que permite diagnosticar y monitorear de forma general el funcionamiento de algunos componentes esenciales del vehículo, de tal forma que se mantenga informado al propietario del automóvil si es pertinente una revisión más minuciosa del mismo. El dispositivo tiene un subsistema de comunicación GSM/GPRS, que realiza la transmisión de los datos adquiridos en el automóvil a través de la red de internet, hasta un servidor. El dispositivo está basado en la red electrónica CAN Bus que incorporan los automóviles modernos fabricados aproximadamente a partir de 2003, que cumplen con la norma internacional ISO 11898, establecida por la industria automotriz, que tiene en cuenta los protocolos de comunicación OBD-II.

ABSTRACT

It is evident the growing of the cities in Colombia, which is forcing people to move longer distances, this increase the sales of cars in the country. It is important to stand up that the accidents has increased too, and many of these incidents, are due to the high speed or malfunction of the cars. This research points to. The construction of a device to diagnosticate and check in general form the function of some essential components of the car, so it can inform the owner if it is time of a general check of the car. This device has a subsystem of GSM/GPRS which sends the acquired data of the car through the internet to a server. The device is based in the electronics network CAN Bus incorporated since 2003, and is part of the international ISO 11898 standar established for the car industries that take into account the protocol of communication OBD-II.

INTRODUCCIÓN

Es importante ofrecer a las personas un dispositivo de fácil instalación para los automóviles, que permita conocer información acerca del mismo, y así ayudar a conductores y propietarios a proteger sus vehículos y no arriesgar la integridad de sus componentes, ni sus vidas, brindándoles la opción de tener un diagnóstico preliminar de fallas confiable, sin importar donde esté, sin verse obligado a llevar su auto a una revisión técnico mecánica, y que sea capaz de obtener información importante suministrada por el vehículo, puede ser por alguna falla que se está presentando en el auto, o simplemente una señal de monitoreo para llevar un registro del desempeño en el trayecto del vehículo.

En el presente trabajo se detalla el diseño y desarrollo de un dispositivo para el monitoreo y diagnóstico de posibles fallas que se pueden presentar en los automóviles basados en la red electrónica CAN (Controller Area Network) Bus, que incorporan los vehículos modernos. El dispositivo es capaz de adquirir y transmitir estos datos hacia cualquier servidor a través de la red celular por medio del sistema global para la comunicación y el servicio general de paquetes vía radio GSM/GPRS (Global System for Mobile communications/General Packet radio Service), que tiene incorporado; también cuenta con un sistema de posicionamiento global GPS (Global Positioning System), que permite saber la ubicación del vehículo. Como sistema central posee un microcontrolador, el cual se encarga de realizar todas las tareas previamente programadas.

En primera instancia, se describe en forma general los diferentes sistemas que componen al dispositivo como el sistema de posicionamiento global y el sistema de comunicación GSM/GPRS. Además, se hace una descripción de la red interna del automóvil llamada CAN Bus y el sistema de diagnóstico de a bordo segunda generación OBD-II (On Board Diagnostics), con su respectivo funcionamiento. Todos estos sistemas estarán controlados por un microcontrolador cuyo diseño es basado en la placa de la plataforma de hardware libre Arduino Duemilanove.

Posteriormente, se muestran los diseños realizados para cada uno de los subsistemas, con la respectiva descripción de cada uno de los componentes utilizados en el desarrollo del dispositivo. Finalmente se presentan las pruebas de funcionamiento realizadas y los resultados obtenidos de la implementación de la tarjeta en un automóvil.

1. DESCRIPCION DEL PROBLEMA

1.1 PLANTEAMIENTO DEL PROBLEMA

Con el pasar del tiempo gran parte de la sociedad colombiana se encuentra interesada en adquirir automóviles, esto se ve reflejado en el aumento de ventas de los mismos y en la congestión vehicular, siendo indispensable mantenerlos en las mejores condiciones para minimizar el riesgo de accidentes en las carreteras del país.

Por lo anterior expuesto, es importante llevar a cabo un monitoreo y registro continuo del funcionamiento de los automóviles en tiempo real, pero los propietarios no tienen un registro confiable del estado del vehículo y se ven en la necesidad de llevar el auto a un centro automotor. Por esta razón, sería mejor que el usuario sea quien realice este proceso de diagnóstico y sepa si su vehículo está en óptimas condiciones y así decidir si es pertinente un diagnóstico más riguroso en un centro especializado. De igual manera es fundamental tener información directa y en línea sobre cualquier vehículo, especialmente del transporte público, vehículos oficiales, ambulancias, entre otros, de tal forma que se puedan detectar a tiempo ciertas fallas, que provocan un mal funcionamiento del vehículo.

1.2 OBJETIVO GENERAL

Diseñar e implementar un dispositivo que sea capaz de adquirir y transmitir datos generados por el automóvil hacia cualquier servidor, para monitorear y diagnosticar posibles fallas que se puedan presentar en el vehículo.

1.3 OBJETIVOS ESPECÍFICOS

1. Analizar la red CAN Bus de los automóviles, en lo referente al protocolo, arquitectura, formato de la trama, mensajes, comandos y opciones.
2. Construir un dispositivo para adquirir los datos más importantes que se desean monitorear en el vehículo.
3. Desarrollar un sistema de comunicación entre el dispositivo ubicado en el automóvil y el computador receptor de los datos.
4. Traducir los datos adquiridos a un archivo de texto, que sea de fácil entendimiento para cualquier otra aplicación.

1.4 JUSTIFICACIÓN

La necesidad de un dispositivo inalámbrico que ayuda al usuario a conocer cualquier tipo de alteración presente en el automóvil, además, el instrumento cuenta con un sistema de posicionamiento global GPS que permite saber la ubicación del vehículo, contribuyendo a incrementar la posibilidad de recuperarlo en caso de robo o cualquier siniestro.

Por otra parte, el dispositivo diseñado permite conocer ciertos estados en los que se encuentra el vehículo y si la conducción es confiable, además, en caso de que ocurra un accidente, la implementación de este hardware contribuirá a determinar las condiciones en las que se encontraba el automóvil al momento de ocurrir el accidente. Se conocerán características como la velocidad a la cual transitaba el auto, la posición del acelerador, y se tendrá la certeza si existió alguna clase de falla en el funcionamiento del sistema del vehículo. Esto puede ser de gran ayuda para diferentes entidades como la policía de tránsito, quienes podrían utilizar el dispositivo para agilizar investigaciones y así poder determinar el grado de culpabilidad de los conductores.

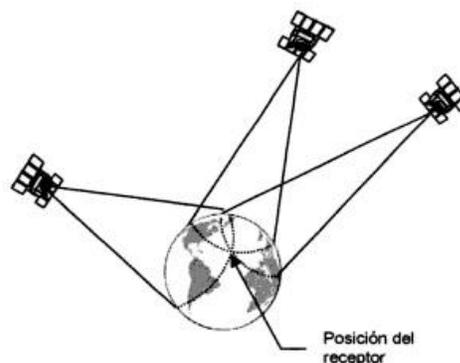
2. MARCO TEORICO

2.1 SISTEMA DE POSICIONAMIENTO GLOBAL (GPS)

El Sistema de Posicionamiento Global (GPS) es un sistema de satélites usado inicialmente en navegación, que permite determinar la posición de vehículos, naves, personas, trenes, entre otros, las 24 horas del día en cualquier lugar del globo terráqueo y en cualquier condición climatológica. Este sistema está compuesto por un conjunto de 24 satélites ubicados aproximadamente a 2000 km de la Tierra, que circundan en la misma y envían señales de radio a su superficie. Un receptor GPS es un aparato electrónico pequeño, permite recibir las señales de los satélites, utilizándolas para calcular su posición [1].

Las señales enviadas por los satélites se mueven a la velocidad de la luz y obviamente como no todos los satélites están a la misma distancia del receptor GPS que se esté usando, llegan a diferentes tiempos. El dispositivo GPS calcula el tiempo que se tarda en llegar la señal desde un satélite, de esa forma sabe a qué distancia se encuentra de él, y hace lo mismo con mínimo cuatro (4) satélites, obteniendo cuatro (4) diferentes distancias. Esto permite “triangular” y determinar con gran exactitud su posición en la Tierra. La Figura 1 muestra un ejemplo de la distribución de los satélites y el cálculo de la posición de un receptor.

Figura 1. Cálculo de la posición de un receptor



Fuente: GPS FÁCIL Uso del Sistema de Posicionamiento Global [1]

Finalmente, el GPS informa sobre su ubicación por medio de coordenadas de latitud, longitud y altitud, mediante trama de datos.

2.2 MICROCONTROLADOR

El microcontrolador es capaz de ejecutar de forma autónoma una serie de instrucciones previamente definidas por el programador, y es el componente principal de la circuitería de procesamiento y control [2].

En el dispositivo ARGUS AUTOMOTIVE LM8095, que es el nombre con el cual se bautizó al instrumento implementado, la conexión del microcontrolador, está basada en la placa de la plataforma de hardware libre Arduino Duemilanove.

2.2.1 Plataforma ARDUINO

Arduino es una placa de hardware libre y de código abierto que incorpora un microcontrolador reprogramable de marca Atmel y posee una serie de pines que permiten conectar de forma sencilla diferentes sensores y actuadores.

Además, la plataforma Arduino tiene un lenguaje propio, basado en C/C++ y por ello soporta las funciones del estándar C y algunas de C++. Sin embargo, es posible utilizar otros lenguajes de programación y aplicaciones populares en Arduino, como Java, Processing, Python, Mathematica, Matlab, Perl, Visual Basic, etc. Esto es posible debido a que Arduino se comunica mediante la transmisión de datos en formato serie, característica que la mayoría de los lenguajes anteriormente nombrados soportan [3].

Cuando se habla de una placa Arduino, se debe especificar un modelo concreto, ya que existen varias placas Arduino oficiales, cada una con diferentes características (como el tamaño físico, el número de pines ofrecidos, el modelo de microcontrolador incorporado, la cantidad de memoria utilizable, etc) [2].

2.3 SISTEMAS DE COMUNICACIÓN

Las comunicaciones móviles hacen referencia a la transmisión de información que se da entre un emisor y un receptor de forma inalámbrica, son sistemas que utilizan comunicación vía radio, por ejemplo comunicación entre aeronaves, vehículos, teléfonos celulares, entre otros, es por eso que si existe algún elemento que requiera procesar datos por medio de un sistema de internet, se clasifica como un sistema de comunicación móvil.

ARGUS AUTOMOTIVE contiene en su esquema un subsistema de comunicación GSM/GPRS, que realiza la transmisión de los datos adquiridos en el automóvil a través de la red de internet hasta un servidor.

2.3.1 Tecnología GSM

El sistema global para comunicaciones móviles GSM es un sistema de telefonía móvil de segunda generación, que permite a sus usuarios conectarse a un computador por medio de un teléfono para enviar y recibir mensajes por correo electrónico y faxes, además de permitir el servicio de mensajes de texto SMS (Short Message Service) y también establecer llamadas de voz. GSM permite llevar a cabo todas estas acciones con un flujo máximo de 9.6 Kbps.

En Europa, Australia, la mayor parte de Asia y en América del Sur, están disponibles los sistemas GSM 900 y GSM 1800. En América del Norte, algunas partes de Asia, y América del Sur, se utiliza GSM 1900. Los números 900, 1800, 1900, se refieren a la banda de frecuencia en MHz utilizada por el estándar GSM [4].

2.3.1.1 Arquitectura de una Red GSM

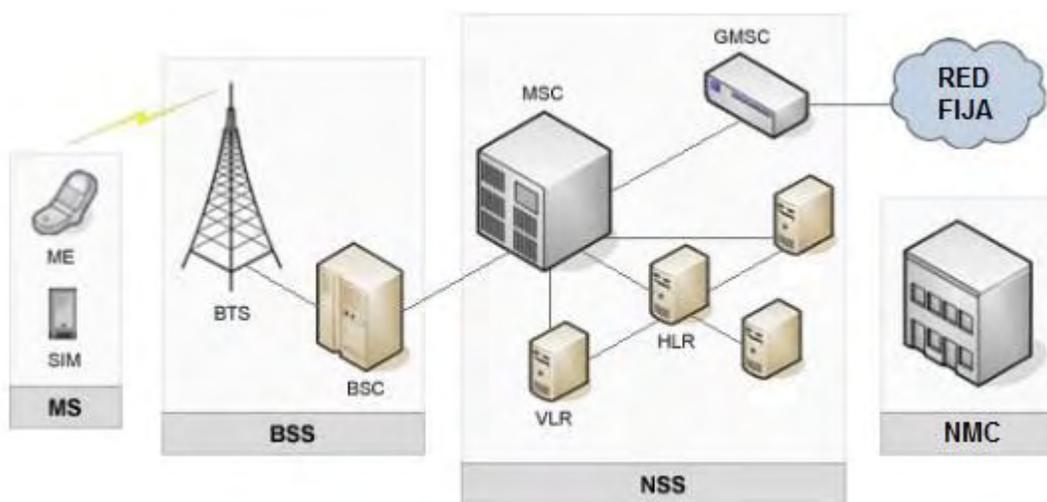
Un sistema GSM se compone de entidades funcionales e interfaces estandarizadas, las cuales permiten adaptar cualquier sistema de red celular con alguna estación móvil. El sistema GSM consta de cuatro partes: la estación móvil MS (Mobile Station), el subsistema de la estación base BSS (Base Station Subsystem), el subsistema de conmutación y red NSS (Network Switching Subsystem) y el centro de gestión de red NMC (Network Management Center) [5].

- Estación Móvil (MS): está compuesta por el equipo móvil y su respectivo módulo de identidad del suscriptor SIM (Subscriber Identity Module), que sirve para identificar al usuario. Una ventaja es que se puede cambiar la SIM entre varios equipos móviles.
- Subsistema de la Estación Base (BSS): controla las comunicaciones de radio del terminal móvil. Está en contacto con el subsistema de red NSS, a través del cual conecta al usuario del móvil con otros usuarios, con ayuda de una estación base que gestiona las comunicaciones de las estaciones móviles, proporcionando un número de canales de radio a la zona que da servicio. Las estaciones bases son controladas para poder mantener las llamadas y adaptar la velocidad del enlace de radio estándar de 64 Kbps [5].
- Subsistema de Conmutación y Red (NSS): está constituido principalmente por 3 elementos: el centro de conmutación de servicios móviles MSC (Mobile Switching Center Services), el registro de ubicación de usuario visitante VLR (Visitor Location Register User) y el registro de ubicación de usuario doméstico HLR (Home User Location Register). El MSC es el responsable de controlar las llamadas en la red móvil, identificando el

origen y el destino de una llamada, así como el tipo de llamada. Un MSC que actúa como puente entre una red móvil y una red fija, es denominado MSC pasarela GMSC (Gateway Mobile Switching centre). El MSC está normalmente junto con el VLR, el cual es una base de datos que mantiene información relacionada a los suscriptores que están en un determinado momento en el área de servicio del MSC. El VLR procede con los registros de ubicación de los MS y la actualización de los mismos. La base de datos del VLR es siempre temporal, en el sentido que la información se mantiene mientras el suscriptor este dentro del área de servicio; en tanto que el HLR mantiene en su base de datos un registro permanente de los suscriptores. En adición a la información fija, el HLR también mantiene una base de datos temporal, la cual contiene la ubicación actualizada de los clientes. Esta información es necesaria para el encaminamiento de las llamadas [4].

- Centro de Gestión de Red (NMC): es el centro de operación de la red, monitorea funciones y componentes de la red móvil, además administra y gestiona posibles fallas de la misma, ya sea de configuración o de desempeño. La Figura 2, muestra la arquitectura completa de la red.

Figura 2. Arquitectura de una red GSM



Fuente: Arquitectura de la red GSM

URL: <http://p.se-todo.com/pravo/11171/index.html?page=2>

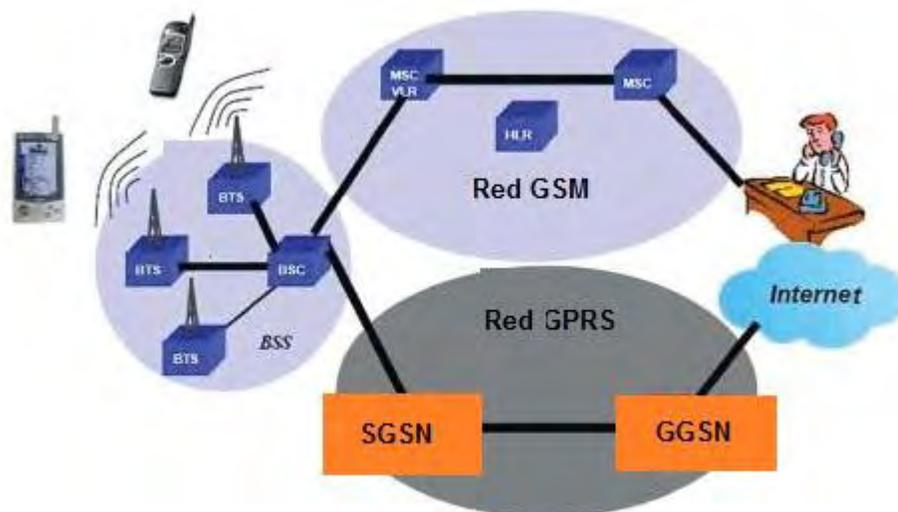
2.3.2 Tecnología GPRS

Servicio general de paquetes vía radio GPRS es una tecnología europea derivada del GSM y corresponde a lo que se conoce como generación 2.5. Con GPRS se consigue beneficiarse de todas las ventajas de la transmisión de datos en movilidad, mejorando la eficiencia, la velocidad y la comodidad de sus comunicaciones. La velocidad máxima teórica de conexión puede llegar a los 115 Kbps, 12 veces más que la permitida por la red GSM [6].

2.3.2.2 Arquitectura de una Red GPRS

GPRS puede transportar paquetes de protocolo de internet IP (Internet Protocol) y comparte la red de acceso de GSM, es decir, las MS y las BSS. Además, GPRS introduce tres nuevos nodos funcionales respecto a GSM, que son el nodo de soporte GPRS SGSN (Serving GPRS Support Node), la puerta de enlace GPRS-Nodo de soporte GGSN (Gateway GPRS Support Node) y la unidad de control de paquetes PCU (Packet Control Unit). El primero, es el responsable de la entrega de paquetes al terminal móvil en su área de servicio, y el segundo actúa como interfaz lógico hacia las redes de paquetes de datos externas. En cambio, el PCU es el encargado de gestionar el interfaz aire de la red [7]. La Figura 3, ilustra la arquitectura de esta red.

Figura 3. Arquitectura de una red GPRS



Fuente: Ingeniería y Desarrollo
URL: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0122-34612009000100011

Entonces, cuando un usuario transmite datos, estos son encapsulados en paquetes cortos, en cuya cabecera se indica las direcciones origen y destino. Cada uno de estos paquetes puede seguir rutas diferentes a través de la red hasta llegar a su destino, así mismo los paquetes originados por distintos usuarios pueden ser intercalados, de esta forma se comparte la capacidad de transmisión. Los paquetes, no son enviados a intervalos de tiempo, sino que cuando es necesario se asigna la capacidad de la red siendo liberada cuando no es necesaria. GPRS utiliza los recursos de radio solamente cuando hay datos que enviar y recibir, adaptándose así perfectamente a la muy intermitente naturaleza de las aplicaciones de datos. Esto permite a los operadores ofrecer un servicio a mejor precio, ya que la facturación se puede basar en la cantidad de datos enviados o recibidos [5].

GPRS aporta servicios como: envío de mensajes cortos, servicios generales de internet, conexiones a intranets, protocolos de aplicaciones inalámbricas WAP (Wireless Application Protocol), servicios específicos para GPRS, y servicios basados en la localización [8].

2.4 COMANDO HAYES O COMANDOS AT

El conjunto de comandos Hayes es un lenguaje desarrollado por la compañía Hayes Communications, que prácticamente se convirtió en estándar abierto de comandos para configurar y parametrizar módems; además, tiene el propósito de utilizar la comunicación serial para hacer el control de aparatos de forma específica, como de algunos módulos GSM/GPRS. Estos comandos también son conocidos como comandos AT, haciendo referencia a “atención”. En dichos comandos si se mide la longitud de los bits, entonces se puede determinar en detalle la velocidad de transmisión [9].

Los comandos Hayes poseen una sintaxis propia [10], y hay tres formas de trabajar comandos AT:

- **Sintaxis básica:** Tienen el formato “AT<x><n>” o “AT&<x><n>” donde x es el comando y n son los argumentos para el comando.
- **Sintaxis de parámetros S:** Tienen el formato “ATS<n>=<m>” donde “<n>” es el índice de los registros S por establecer, y “<m>” es el valor que se

asigna a la misma “m” (es opcional). Si no se pone valor, entonces se establece un valor por defecto.

- **Sintaxis extendida:** Estos comandos pueden funcionar de diferentes maneras, como muestra la Tabla 1.

Tabla 1. Sintaxis extendida de comandos AT

Título	Comando	Descripción
Comando de prueba	AT+<x>=?	Devuelve la lista de parámetros y valores establecidos.
Comando de lectura	AT+<x>?	Devuelve el valor actual del parámetro o parámetros.
Comando de escritura	AT+<x>=<...>	Establece los valores de los parámetros que define el usuario.
Comando de ejecución	AT+<x>	Lee parámetros que no varían, afectados por procesos internos del módulo GSM.

Fuente: Manual comandos AT

URL:

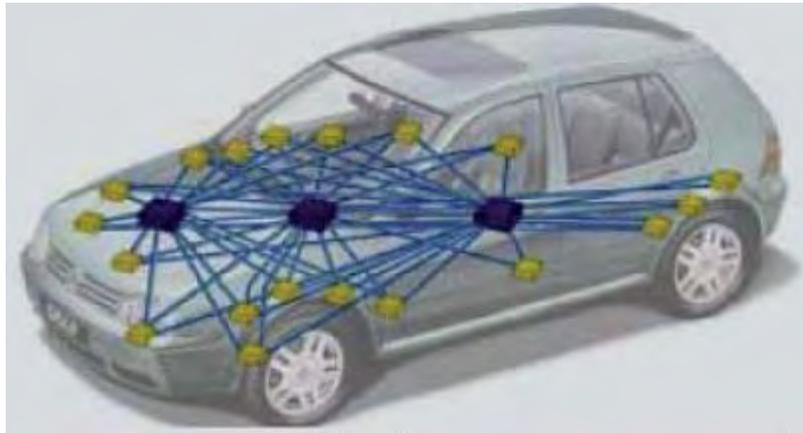
http://www.quectel.com/UploadImage/Downlad/M95_AT_Commands_Manual_V1.2.pdf

2.5 PROTOCOLO CAN BUS

El protocolo de comunicación CAN se basa en una arquitectura de tipo bus, que sirve para la transmisión de mensajes en tiempo real con los diferentes dispositivos conectados en esta red como sensores, controladores, transmisores, entre otros. Además, tiene un alto nivel de confiabilidad, eficiencia y rapidez, lo cual hizo que las compañías automotrices implementaran este sistema de comunicación, convirtiéndose así en una de las redes vehiculares más usadas en la actualidad la industria automotriz, interconectar todos los sistemas electrónicos de un carro implicaba un amplio y extenso cableado, lo que conllevaba a que los automóviles fuesen más pesados y en caso de haber una deficiencia en el funcionamiento del mismo, era muy difícil encontrar el error. En el ANEXO 1 se describe la reseña histórica de la red CAN Bus.

En un automóvil existen varias unidades de control, las cuales tienen la necesidad de compartir información entre sí para el correcto funcionamiento del vehículo. Entre ellos están las unidades de control del motor, unidades de control de la transmisión, unidades de control del sistema antibloqueo de ruedas ABS (Antilock Brake System), unidades de control de cambios, unidad de control de dirección, entre otras. En la Figura 4 se observa como es el cableado de conexión entre tres de estas unidades de control en un vehículo, antes de la implementación del sistema CAN Bus.

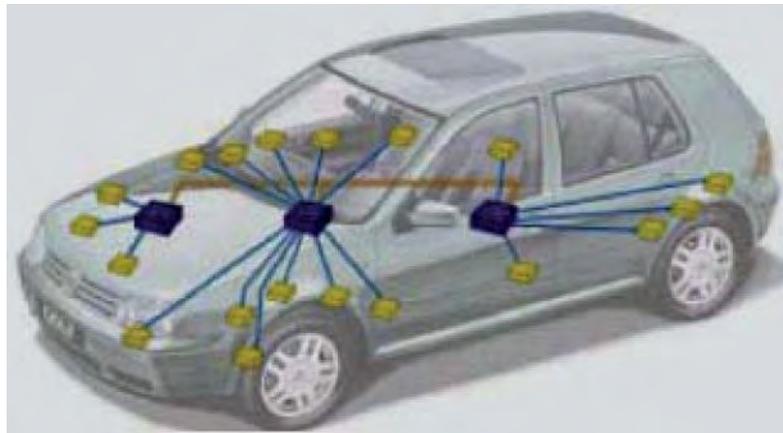
Figura 4. Conexión unidades de control sin CAN Bus



Fuente: Compendio del sistema de Transmisión de datos CAN Bus [11]

Las exigencias de los clientes en cuanto a los automóviles crecen con el tiempo, ya sea para mayores comodidades en cuanto al confort del vehículo, seguridad del mismo e incluso consumo de combustible. Esto motivó al uso del CAN Bus en los automóviles, reduciendo notoriamente su cableado interno. En la Figura 5, se muestra el cableado del vehículo usando red CAN Bus.

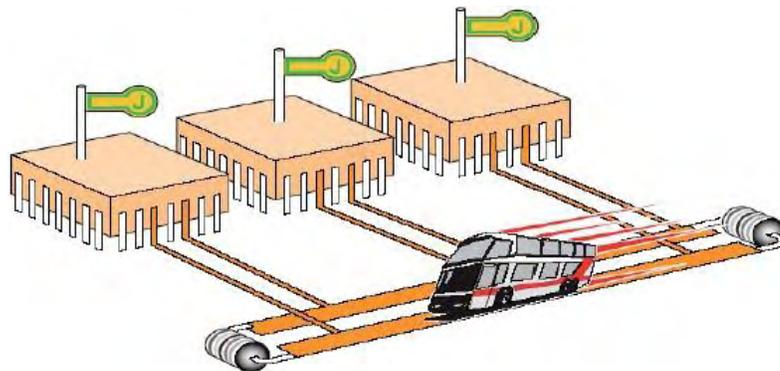
Figura 5. Conexión entre unidades de control con CAN Bus



Fuente: Compendio del sistema de Transmisión de datos CAN Bus [11]

La función del CAN Bus en el automóvil es interconectar diferentes unidades de mando entre sí, también llamados módulos o nodos, de forma que pueden intercambiar información en un tiempo muy reducido, al igual que un autobús puede transportar un gran número de personas, el CAN Bus puede trasportar una gran cantidad de información [11], como se ilustra en la Figura 6.

Figura 6. Esquema de comparación del CAN Bus con el medio físico



Fuente: Compendio del sistema de Transmisión de datos CAN Bus [11]

Con la implementación de este protocolo se dio gran solución al problema de eficiencia con respecto al funcionamiento de las unidades de control, debido al número elevado de conexiones en el interior del automóvil. CAN hace una mejor distribución del sistema electrónico facilitando su mantenimiento y manipulación, además que el montaje y desmontaje se realiza en un menor tiempo reduciendo algunos costos.

2.5.1 Características Básicas Del CAN BUS

CAN es un protocolo orientado a mensajes que está basado en identificadores. A cada mensaje se le asigna un identificador único, y se lo encapsula en tramas para ser transmitido por medio de la red hacia todos los módulos de forma

simultánea. Cuando el mensaje llega a los módulos, éstos en primera instancia analizan el identificador para decidir si dicha información les concierne o por el contrario deben rechazar el mensaje. Además, CAN es un protocolo de comunicación serie que soporta control distribuido en tiempo real con un alto nivel de seguridad y multiplexación ya que todos los módulos están conectados entre sí a través de un bus, lo cual hace que puedan compartir cualquier tipo de información entre ellos, garantizando que los tiempos de comunicación sean mínimos.

2.5.1.1 Normas de Estandarización

Se trata de un estándar determinado por las normas definidas por la organización internacional de estandarización ISO (International Organization for Standardization), específicamente en la ISO 11898, que ha sido modificada a través del tiempo y dividida en diferentes estándares aplicables a CAN [12]:

- **ISO 11898 – 1:** Capa de enlace y señalización física.
- **ISO 11898 – 2:** Unidad de acceso al medio de alta velocidad.
- **ISO 11898 – 3:** Interfaz de baja velocidad tolerante a fallos.
- **ISO 11898 – 4:** Comunicación: tiempo activado para la comunicación Time-Triggered”.

2.5.1.2 Estructura Definida

La información que circula entre los módulos a través del bus son paquetes de bits con una longitud limitada y una estructura definida de campos que conforman el mensaje.

2.5.1.3 Número de Nodos

Es posible conectar hasta 32 nodos en una sola red CAN. Esto es muy importante porque brinda a la red flexibilidad y posibilidad de expansión, ya que nuevos nodos pueden ser conectados sin necesidad de hacer ninguna modificación en su hardware ni software [13].

2.5.1.4 Optimización del Ancho de Banda

Todos los métodos que se utilizan para la distribución de mensajes a través del bus, como por ejemplo la prioridad en el envío de cada uno de estos según su identificador, contribuyen a un mejor aprovechamiento de ancho de banda disponible.

2.5.1.5 Detección y Desactivación de Nodos Defectuosos

El protocolo CAN Bus monitorea funciones específicas en la comunicación de nodos, de tal manera que un nodo defectuoso no pueda afectar continuamente la transmisión de datos. Cuando en un nodo se excede la tasa promedio de error

predefinida, se limita el acceso del nodo afectado a la red, o se lo desconecta de la misma [14].

2.5.1.6 Relación Velocidad/Distancia

La velocidad depende de la distancia hasta un máximo de 1000 metros, aunque es posible usar repetidores o puentes de red. La Tabla 2 muestra la comparación entre la velocidad y distancia para el protocolo CAN Bus [13]:

Tabla 2. Relación velocidad/distancia para el protocolo CAN Bus

Velocidad (Kbps)	Tiempo de bit (us)	Longitud máxima de bus (m)
1000	1	30
800	1.25	50
500	2	100
250	4	250
125	8	500
50	20	1000
20	50	2500
10	100	5000

Fuente: Diseño e Implementación de Protocolo CAN para el control de un Módulo de Red de Sensores [13]

2.5.1.7 Recepción por Multidifusión (Multicast)

Permite que todos los nodos asociados a la red puedan recibir el mismo mensaje al mismo tiempo, además tienen la capacidad de acceder al bus de forma

simultánea con sincronización de tiempos, realizando simultáneamente un proceso de filtrado.

2.5.1.8 Jerarquía Multimaestro

Este sistema multimaestro permite que dado un mismo tiempo pueda haber más de un maestro en la red, es decir que cada módulo puede iniciar su transmisión tan pronto como el bus esté desocupado. Esto permite que los sistemas sean inteligentes y redundantes.

2.5.2 Estructura de Capas del Protocolo CAN Bus

El protocolo de comunicaciones CAN, emplea el modelo de interconexión de sistemas abiertos OSI (Open Systems Interconnection), el cual presenta una arquitectura que cubre de forma estandarizada la capa física y de enlace, garantizando transparencia en el diseño y flexibilidad en las implementaciones. Además, bajo la especificación de la norma ISO 11898, se describe cómo la información será transmitida entre los diferentes módulos de la red. La Tabla 3 indica la arquitectura de las capas del protocolo CAN Bus bajo las normas ISO 11898.

Tabla 3. Arquitectura de las capas del protocolo CAN Bus

Capa Enlace de Datos	
LLC (Logical Link Control)	<ul style="list-style-type: none"> • Filtrado de mensajes. • Notificación de Sobrecarga. • Proceso de Recuperación.
MAC (Medium Access Control)	<ul style="list-style-type: none"> • Encapsulamiento / Des encapsulamiento de datos. • Codificación de tramas (bits de relleno). • Arbitrar el acceso al medio.
Capa Física	
<ul style="list-style-type: none"> • Medio Físico • Niveles electrónicos de transmisión • Elementos del sistema CAN Bus 	

Fuente: Estudio para Simular una Red CAN con aplicación en dispositivos electrónicos en el automóvil [12]

En el ANEXO 2 se describe de forma detallada, la arquitectura de cada una de las capas del protocolo CAN Bus.

2.5.3 Tramas Protocolo CAN Bus

El protocolo CAN tiene dos formatos para transmitir datos: el formato de trama estándar, según la especificación CAN 2.0 A y el formato de trama extendida, según la especificación CAN 2.0 B. La principal diferencia entre las dos, es la longitud del identificador de mensaje (ID). En el caso de la trama estándar, soporta una longitud de 11 bits (2032 identificadores), y para el caso de la trama extendida soporta una longitud de 29 bits (más de 536 millones de identificadores).

2.5.3.1 Tipos de Tramas

Existen distintos tipos de tramas predefinidas por CAN para la gestión de la transferencia de mensajes [15]:

- Trama de datos.
- Trama de información remota.
- Trama de error.
- Trama de sobrecarga.

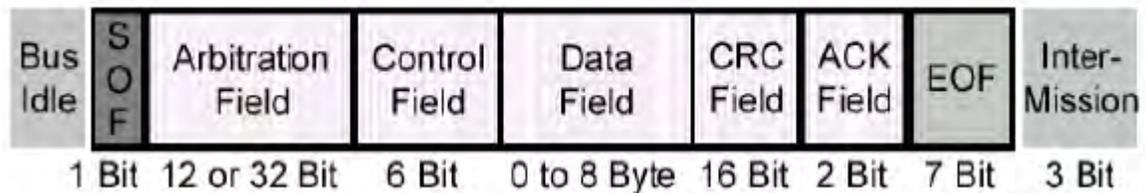
En un bus CAN los nodos transmiten la información espontáneamente con tramas de datos, bien sea por un proceso cíclico o activado ante eventos en el nodo. La trama de información remota sólo se suele utilizar para detección de presencia de nodos o para puesta al día de información en un nodo recién incorporado a la red.

La trama con la cual se trabaja en el proyecto es la trama de datos, a continuación se hace una breve descripción de ella, y el estudio de los otros tipos de trama se presentan en el ANEXO 3.

- **Trama de datos**

Es la información útil que se transmite en “broadcast” a todos los demás nodos, puede incluir entre 0 y 8 Bytes o (0 a 64 bits) de información. Un formato de trama estándar está formado por los siguientes campos de bits: inicio de trama, campo de arbitraje, campo de control, campo de datos, campo de aseguramiento, campo de confirmación, campo de final de trama, y el espacio entre tramas como se ve en la Figura 7 [15].

Figura 7. Formato de trama



Fuente: BUS CAN

URL: <http://server-die.alc.upv.es/asignaturas/PAEEES/2005-06/A03-A04%20-%20Bus%20CAN.pdf>

- **Inicio de trama SOF (Start Of Frame):** es una celda de un solo bit siempre dominante, que indica el inicio de la transmisión, y su flanco descendente sirve para la sincronización con otros nodos.
- **Campo de arbitraje (Arbitraje Field):** en total está formado por 12 bits, los 11 primeros bits de este campo son empleados como identificador que permite reconocer a las unidades de control la prioridad del mensaje. Entre más bajo sea el valor del identificador más alta será la prioridad del mensaje. El último bit RTR (Remote Transmission Request), se utiliza para indicar si el mensaje a transmitir es una trama de datos (RTR=0 dominante), o una trama remota sin datos (RTR=1 recesivo). Cabe destacar que una trama de datos tiene una prioridad más alta que una trama remota.
- **Campo de control (Control Field):** lo componen 6 bits; 1 bit de identificador extendido IDE (Identifier Extension) utilizado para indicar si la trama es estándar (IDE=0 dominante) o si es extendida (IDE=1 recesiva). El segundo bit RBO siempre es recesivo y se usa para futuras expansiones del sistema y los siguientes 4 bits de código de longitud de datos DLC (Data Length Code) indican el número de bytes a transmitir en el campo de datos.
- **Campo de datos (Data Field):** muestra información del mensaje con los datos que el módulo correspondiente introduce en la línea del CAN Bus. Su tamaño varía entre 0 a 8 bytes, es decir 0 a 64 bits en saltos de 8.
- **Campo de chequeo de errores CRC (Cyclic Redundant Code):** es un código de redundancia cíclica que genera el transmisor por la división

módulo 2 de todos los bits precedentes del mensaje, incluyendo los de relleno si existen. Se da a través del polinomio generador: $X^{15} + X^{14} + X^8 + X^7 + X^4 + X^3 + X + 1$, y el resto de esta división es el código CRC transmitido. Los receptores comprueban este código. Este campo tiene una longitud de 16 bits, los 15 primeros son utilizados para la detección de errores, mientras que el último siempre será un bit recesivo (1) para delimitar el campo.

- **Campo de reconocimiento ACK (Acknowledge Field):** formado por 2 bits ACK-slot y ACK-delimiter, que son siempre transmitidos como recesivos (1). Todos los módulos que reciben el mismo CRC modifican el primer bit del campo ACK por un dominante (0), de esta forma el módulo que todavía está transmitiendo, reconoce que al menos un módulo ha recibido un mensaje correctamente escrito. De no ser así, la unidad de mando transmisora interpreta que su mensaje presenta un error.
- **Campo de final de mensaje OEF (End Of Frame):** este campo indica el final del mensaje con una cadena de 7 bits recesivos.
- **Espacio entre tramas IFS:** consta de un mínimo de tres bits recesivos.

2.5.3.2 Validación de Tramas

- Transmisor: La trama es válida si no existen errores hasta el final del campo EOF.
- Receptor: La trama es válida si no existen errores hasta el siguiente bit después del campo EOF [13].

2.5.4 Proceso para la Transmisión de Datos

Para el proceso de transmisión de datos se deben realizar los siguientes pasos:

2.5.4.1 Suministro de Datos

Los módulos reciben la información de los sensores asociados a cada uno de ellos, para el desarrollo del proyecto serán los que ayudan a determinar las revoluciones por minuto, temperatura del líquido refrigerante, la velocidad del automóvil, la posición del acelerador, y el porcentaje de carga del motor. Después, el microprocesador que se encuentra ubicado en la parte interior de cada módulo es el encargado de pasar la información hacia el controlador donde es preparada para, a su vez, ser pasada al transceptor quien la traducirá en señales eléctricas.

2.5.4.2 Transmisión de Datos

Para este proceso el controlador es el encargado de la transferencia de datos y el identificador, junto con la petición de inicio de transmisión, asumiendo la responsabilidad para que el mensaje sea correctamente transmitido a todos los módulos de la red [16].

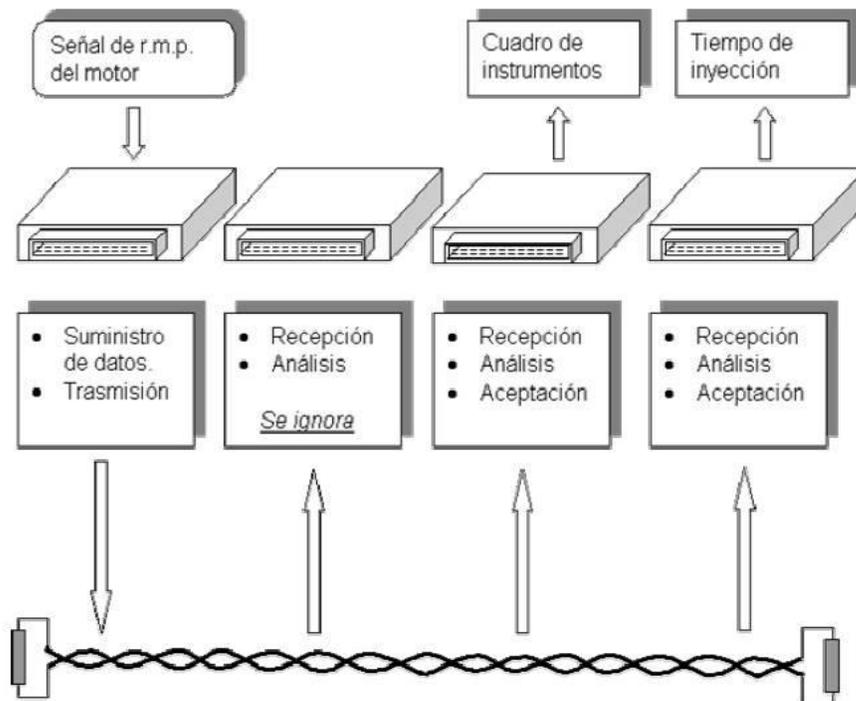
Para comenzar a transmitir un mensaje el módulo tiene que encontrar el bus libre, y dado el caso que exista algún conflicto con otro módulo al intentar transmitir al mismo tiempo, este problema es resuelto por el arbitraje utilizando el identificador. El mecanismo de arbitraje garantiza que ni la información ni el tiempo se pierdan.

2.5.4.3 Recepción de Datos

Cuando todos los módulos reciben el mensaje, verifican el identificador para determinar si el mensaje va ser utilizado por ellos. Los módulos que necesiten los datos del mensaje lo procesan, mientras que los que no lo necesitan lo ignoran [16].

A continuación en la Figura 8 se puede observar el proceso de transmisión de datos en el automóvil.

Figura 8. Proceso de transmisión de datos en el automóvil



Fuente: Nodo de comunicación basado en el Bus CAN [16]

2.6 SISTEMA DE DIAGNÓSTICO OBD-II

El OBD-II (Sistema de Diagnóstico de A Bordo Segunda Generación), en la actualidad es totalmente asentado como método para contrarrestar la producción de emisiones, para evitar la contaminación atmosférica y así preservar el medio ambiente.

En el año de 1989 comenzaron los estudios acerca del OBD-II y posteriormente fue implantado inicialmente en California en 1994. Solamente a partir de 1996 la norma fue adoptada en todos los Estados Unidos, a partir de esta fecha los vehículos fabricados e importados por los Estados Unidos tendrían que cumplir con esta norma. En Latinoamérica esa norma aparece en vehículos de una forma muy complicada ya que hay vehículos importados de Estados Unidos sin tener OBD-II, aun teniendo el conector normalizado [17].

El sistema OBD-II es una mejora al sistema OBD. Es un sistema de diagnóstico electrónico que se encuentra integrado dentro del vehículo, capaz de detectar efectivamente una falla en el sistema de inyección con el objetivo principal de reducir las emisiones de gases contaminantes producidas por los vehículos. Además, tiene múltiples ventajas en cuanto al rendimiento del motor debido a las mejoras continuas dentro de su configuración. Cuando el sistema del automóvil presenta alguna falla en su funcionamiento, las emisiones de gases aumentan fuera de los límites, y es ahí donde el OBD-II actúa para informar sobre la detección de alguna falla. Cabe destacar que los fallos producidos en el vehículo primero se cargan en una memoria antes de ser informados al usuario.

El funcionamiento del OBD-II se da a través de un conjunto de instrucciones programadas en la computadora a bordo del vehículo. Los programas están diseñados específicamente para detectar fallas en los sensores, accionadores, y el cableado de los distintos sistemas del vehículo relacionado con las emisiones. Si la computadora detecta alguna falla en cualquiera de estos componentes o sistemas, realiza un chequeo de lazo cerrado durante un intervalo de tiempo con el fin de saber si el daño es temporal o permanente. Cuando se encuentra que el daño es permanente, la computadora enciende un indicador en el panel de instrumentos llamada Check Engine, a fin de alertar al conductor. El indicador se ilumina solamente cuando se encuentra un problema relacionado con las emisiones. En la Figura 9 se puede observar la ubicación del Check Engine.

Figura 9. Ubicación indicador Check Engine en automóviles



La computadora también asigna un código numérico para cada problema específico que detecta e identifica donde ocurrió la falla, y almacena estos códigos en su memoria para recuperarlos más tarde. A estos códigos se puede acceder a través del puerto OBD-II universal, que suele ubicarse debajo del panel de instrumentos, aproximadamente en un radio de 30cm alrededor, tal como muestra la Figura 10.

Figura 10. Ubicación del conector OBD-II en el Automóvil



2.6.1 Modos de Prueba del OBD-II

- Modo 1: sirve para la identificación del parámetro (PID). A través de este modo se puede tener acceso a los datos analógicos o digitales de salidas y entradas de la computadora central en tiempo real. Este modo también es conocido como flujo de datos.
- Modo 2: se tiene acceso al cuadro de datos congelados. Esta función es muy importante para el OBD-II porque la computadora central toma una muestra de todos los valores relacionados con las emisiones de gases, al momento de ocurrir cualquier falla. De esta forma, al recuperar estos datos se puede conocer las condiciones exactas en las que ocurrió dicha falla.
- Modo 3: este modo permite leer de la memoria de la computadora central todos los códigos de falla DTC (Data Trouble Code).
- Modo 4: a través de este modo se pueden eliminar todos los datos guardados incluyendo los códigos de fallas y el cuadro de datos congelados.
- Modo 5: este modo devuelve todos los datos de las pruebas realizadas a los sensores de oxígenos, esto con el fin de determinar el funcionamiento de los mismos y la eficiencia del convertidor catalítico.
- Modo 6: permite obtener los resultados de todas las pruebas de abordó.
- Modo 7: este modo permite leer de la memoria de la computadora central todos los DTCs pendientes.
- Modo 8: al seleccionar este modo se puede realizar una prueba a los actuadores, y así mismo el mecánico podrá activar y desactivar dichos actuadores.

2.6.2 Códigos de Diagnóstico Fallas DTC

Los modos de prueba del diagnóstico OBD-II han sido creados de tal forma que sean comunes para todos los vehículos sin importar el fabricante. De esta manera con la tarjeta realizada no importa la clase de vehículo a la cual se la conecte, y únicamente se debe tener en cuenta que el vehículo cuente con el estándar OBD-II.

La Sociedad Americana de Ingenieros SAE (Society of Automotive Engineers) publicó la norma J2012 para estandarizar el formato de los códigos de diagnóstico. Este formato permite que los scanner genéricos accedan a cualquier sistema. El

formato asigna códigos alfanuméricos a las fallas y cada uno de los dígitos presenta una ruta específica del diagnóstico, como especifica la Figura 11. Esto ayuda rápidamente a determinar el sistema afectado, además de proveer una guía de mensajes asociados con estos códigos. Por otra parte, existen algunas fallas que no tiene un código asignado, esto hace que sean propias de cada fabricante de vehículos.

Figura 11. Formato de códigos de falla



2.6.3 Conector de Diagnóstico Estándar

El conector de diagnóstico permite la interacción con la información del vehículo. Este es estandarizado para permitir el uso del escáner original de la marca de los vehículos o cualquier otro tipo de escáner genérico. Los pines del conector OBD-II tienen funciones asignadas, las cuales corresponden a diferentes normativas y protocolos de comunicación de acuerdo a los requerimientos de cada fabricante. A continuación la Tabla 4 ilustra una lista que describe la función de cada pin, de acuerdo a una de las configuraciones más usadas por los fabricantes de automoviles.

Tabla 4.Descripción de pines del Conector OBD-II

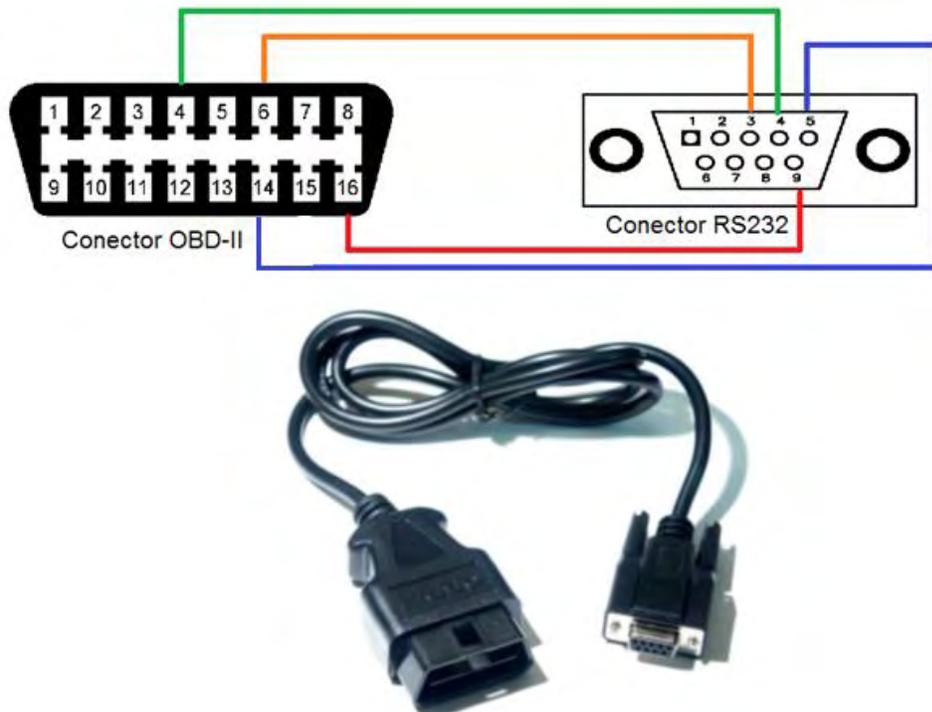
PIN	Función
1	Uso del fabricante
2	J 1850 (Bus +)
3	Uso del fabricante
4	Tierra (chasis)
5	Señal de tierra
6	Bus de datos Can High
7	Línea K ISO 9141-2
8, 9	Uso del fabricante
10	J 1850 (Bus -)
11, 12, 13	Uso del fabricante
14	Bus de datos Can Low
15	Línea L ISO 9141-2
16	Voltaje de batería

Según la SAE, se estableció que las líneas de Can_H y Can_L salgan al exterior por medio del conector estandar OBD-II a través de los terminales 6 y 14 respectivamente. Esta norma está vigente desde el año 2008.

2.6.4 Cable Conector Entre Puerto OBD-II y Tarjeta ARGUS AUTOMOTIVE

Este tipo de cable tiene en uno de sus extremos un conector estándar de diagnóstico OBD-II y en su otro extremo un conector RS232. En la conexión interna se observan cuatro cables, uno de ellos es la conexión de la tierra del carro con la tierra de la tarjeta, que se da a través del pin 4 del conector OBD-II con el pin 2 del RS232, el segundo cable es para la alimentación de la tarjeta desde la batería del carro, que se da por medio de la conexión del pin 16 del conector OBD-II con el pin 9 del RS232. Por último la conexión del sistema CAN Bus del automovil que se da a través del pin 6 y pin 14 del conector OBD-II, cada uno de estos representa las líneas de Can_H y Can_L respectivamente que se unen al pin 3 y pin 5 del conector RS232. En la Figura 12 se puede observar la conexión interna del cable conector.

Figura 12. Cable conector con terminales OBD-II y RS232



Actualmente se está desarrollando la planeación de OBD-III, que toma de base a OBD-II pero da un paso hacia la comunicación de fallas a distancia vía satélite, utilizando un pequeño radio comunicador. Un vehículo equipado con el sistema OBD-III tiene la capacidad de reportar problemas de emisiones directamente a una agencia reguladora de emisiones. El radio comunicador podrá informar el tipo de vehículo y podrá diagnosticar códigos que estén presentes [18].

3. DISEÑO DEL DISPOSITIVO ARGUS AUTOMOTIVE LM8095

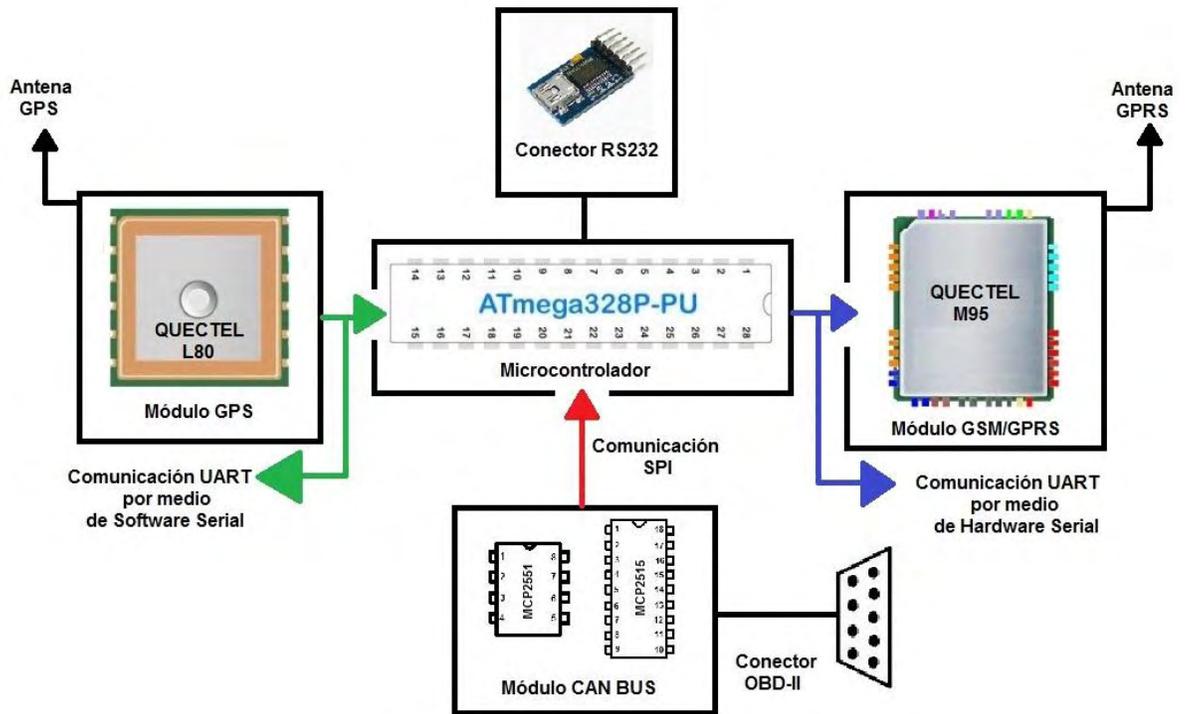
ARGUS AUTOMOTIVE está controlado por el microcontrolador ATmega328P-PU, siendo este elemento el cerebro del dispositivo. Todos los módulos que componen la tarjeta diseñada, se deben conectar y comunicar con el microcontrolador.

La parte del diseño que se refiere a la adquisición de datos del automóvil por medio de CAN Bus, se conecta al microcontrolador a través de comunicación periférica serie SPI (Serial Peripheral interface). La parte de transmisión de datos desde el carro hasta un servidor, que se hace por medio del dispositivo GSM/GPRS Quectel M95, se comunica con el ATmega328P-PU con comunicación UART (Universal Asynchronous Receiver-Transmitter), gracias al puerto serial que posee el microcontrolador.

Además, el posicionamiento GPS del instrumento se logra con el módulo Quectel L80, que también se comunica con el microcontrolador por medio de comunicación UART a través del puerto serial. Sin embargo, el microcontrolador ATmega328P-PU sólo posee un puerto serial, el cual es usado en este proyecto por el módulo GSM/GPRS. Es por eso que se debe crear por medio de software un nuevo puerto en el microcontrolador, utilizando la biblioteca de Arduino Software Serial, que brinda la posibilidad de permitir la comunicación en serie en otros pines digitales del ATmega328P-PU. Es decir, se puede crear tantos puertos seriales como el microcontrolador permita.

Finalmente, la conexión del dispositivo con un computador para su respectiva programación se hace por medio del integrado RS232, La Figura 13 muestra el diagrama de bloques del esquema del dispositivo.

Figura 13. Diagrama de bloques de ARGUS AUTOMOTIVE



3.1 MICROCONTROLADOR ATMEGA328P-PU

El ATmega328P-PU fue seleccionado para el diseño del proyecto debido a su excelente relación costo/eficiencia, además de tener un número de entradas y salidas digitales que se ajustan a las necesidades del dispositivo final. También permite trabajar con diferentes tipos de comunicación como UART, SPI y circuito inter-integrado I2C (Inter-Integrated Circuit). Las características más relevantes que ofrece el microcontrolador ATmega328P-PU, se muestran en la Tabla 5 y el ANEXO 4 identifica los pines del microcontrolador.

Tabla 5. Características del microcontrolador ATmega328P-PU

Parámetros ATmega328P-PU	Valores
Voltaje de operación	5V
Pines digitales	14
Pines digitales PWM	6
Pines de entradas analógicas	6
Memoria flash (Kbytes)	32 KB
SRAM	2 KB
EEPROM	1 KB
Puertos seriales	1
Comunicación SPI	2
Comunicación I2C	1

Fuente: Datasheet ATmega328P-PU

URL: http://www.atmel.com/images/atmel-8271-8-bit-avr-microcontroller-atmega48a-48pa-88a-88pa-168a-168pa-328-328p_datasheet_summary.pdf

3.1.1 Esquema de Conexión del Microcontrolador ATmega328P-PU

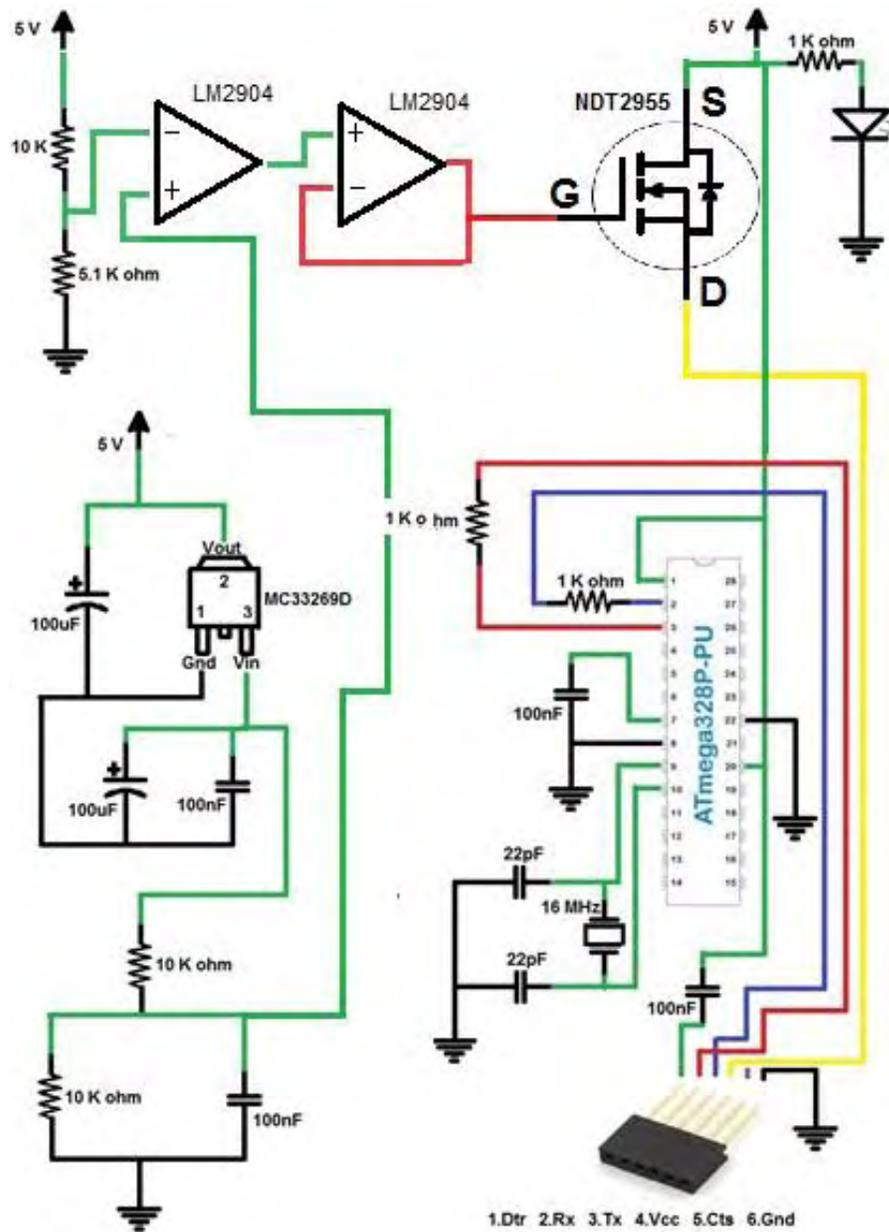
En el dispositivo diseñado, el microcontrolador ATmega328P-PU se interconecta con todos los componentes:

- La adquisición de datos del CAN Bus.
- La transmisión de información por medio del módulo Quectel M95.
- El posicionamiento del instrumento con el receptor GPS Quectel L80.

Además, el desarrollo de la conexión de este microcontrolador se basó en la placa Arduino Duemilanove, porque está compuesta de elementos económicos y fáciles de encontrar en el comercio electrónico. Además, incorpora un conector bus universal en serie USB (universal Serial Bus) instalado, sistemas de protección y fácil accesibilidad a todos los puertos del microcontrolador.

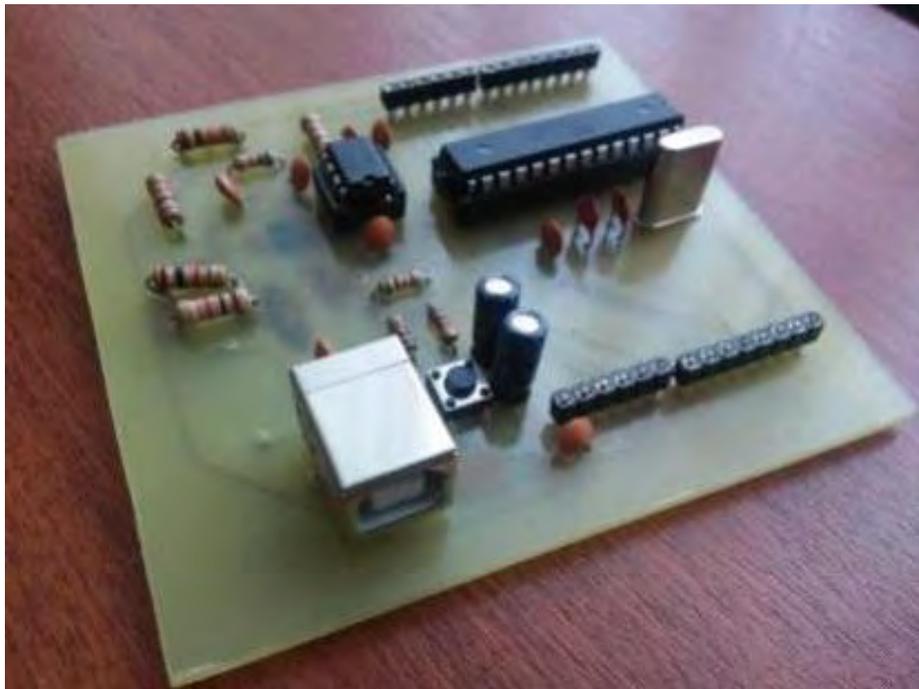
El montaje óptimo para que el microcontrolador funcione, es logrado con etapas de potencia para regular voltajes necesarios y etapas de protección. Se utiliza el componente MC33269D como regulador de 5 voltios, que tiene la característica de servir como protector al existir cualquier subida de tensión. También resalta en el esquema el circuito integrado LM2904, este es un amplificador operacional, con el cual se realizan dos configuraciones, la primera en modo comparador entre el voltaje de alimentación de la tarjeta el cual se hace a través de la batería del automóvil y un voltaje de referencia con valor de 3.3 v, la salida de este amplificador se conecta a una segunda etapa en configuración seguidor de tensión, que no altera el voltaje de entrada pero sirve para acoplar impedancias. Otro elemento que se utiliza es el transistor MOSFET NDT2955 el cual actúa como un conmutador entre el voltaje procedente del segundo amplificador y el voltaje USB que se genera a través del conector FT232RL. Por último, cabe mencionar que el microcontrolador funciona con un oscilador de 16 MHz. La Figura 14 muestra el esquema realizado para el funcionamiento del ATmega.

Figura 14. Esquema de conexión del ATmega328P-PU



En el esquema anterior se puede observar pines de conexión tipo hembra (Dtr, Rx, Tx, Vcc, Cts, Gnd), los cuales facilitan la programación del microcontrolador. Lo que permite que el dispositivo se pueda conectar a cualquier computador es el integrado FT232RL, que es el encargado de realizar la conversión serial / USB; este circuito integrado no es parte del esquema, debido a parámetros del diseño. La Figura 15 muestra la primera fabricación del circuito para el ATmega328P-PU.

Figura 15. Primera fabricación del circuito para el ATmega328P-PU

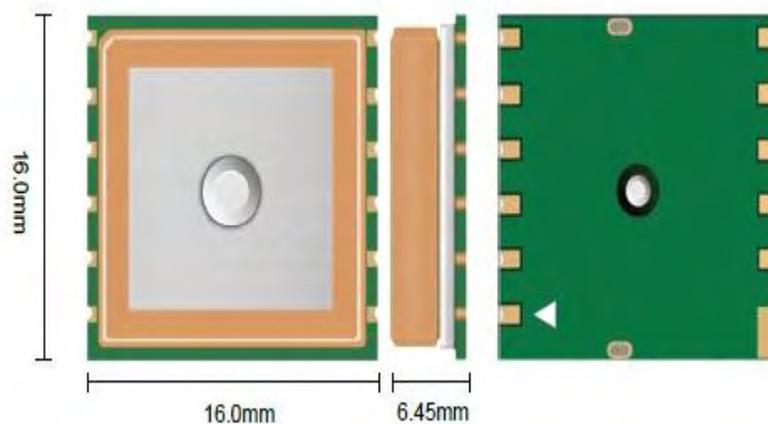


3.2 MÓDULO GPS QUECTEL L80

El dispositivo Quectel L80 es el módulo receptor GPS seleccionado para el desarrollo de este proyecto, ya que posee un diseño compacto y un rendimiento adecuado en la adquisición y seguimiento. Esto permite fijar la posición de forma rápida, incluso a niveles de señal en interiores, haciéndolo con un bajo consumo de energía.

Además de tener una antena de parche incorporada, permite la conexión de una antena externa para mejorar la precisión del dispositivo, contando con 66 canales de búsqueda y 22 canales simultáneos de seguimiento. Así adquiere y rastrea satélites en el menor tiempo posible. La Figura 16 muestra las dimensiones del módulo Quectel L80.

Figura 16. Dimensiones del módulo Quectel L80



Fuente: Datasheet Quectel L80

URL:

http://www.quectel.com/UploadFile/Product/Quectel_L80_GPS_Specification_V1.2.pdf

Una de sus principales características es su bajo consumo de energía, teniendo un voltaje de alimentación entre 3V y 4.3V. Posee un buen rango de trabajo en condiciones ambientales extremas, además trabaja con interfaz serial receptor-transmisor asíncrono universal UART, ajustable entre 4800 y 115200 bps. La Tabla 6 muestra las características del módulo.

Tabla 6. Características del módulo Quectel L80

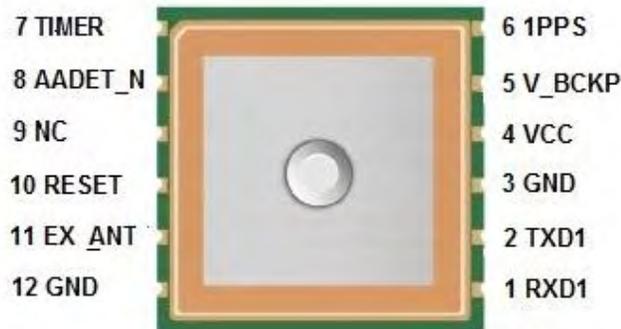
Característica	Implementación
Fuente de alimentación	<ul style="list-style-type: none"> • Voltaje entre: 3.0V – 4.3V
Consumo de energía	<ul style="list-style-type: none"> • Adquisición: 25 mA • Rastreo: 20 mA
Tipo de receptor	<ul style="list-style-type: none"> • Banda de recepción: 1575.42MHz • 66 canales de búsqueda. • 22 canales simultáneos de seguimiento.
Sensibilidad	<ul style="list-style-type: none"> • Adquisición: -148 dBm • Re-adquisición: -160 dBm • Rastreo: -165 dBm
TTFF(Tiempo para la primera trama); con EASY habilitado	<ul style="list-style-type: none"> • Arranque en frío: 15 segundos. • Arranque en normal: 5 segundos. • Arranque en caliente: 1 segundo.
TTFF con EASY deshabilitado	<ul style="list-style-type: none"> • Arranque en frío: 35 segundos. • Arranque en normal: 30 segundos. • Arranque en caliente: 1 segundo.
Exactitud de la posición horizontal	<ul style="list-style-type: none"> • < 2.5 m
Máxima velocidad de actualización	<ul style="list-style-type: none"> • Hasta 10 Hz, 1 Hz por defecto.
Velocidad de la precisión	<ul style="list-style-type: none"> • < 0.1 m/s
Aceleración de la precisión	<ul style="list-style-type: none"> • 0.1 m/s²
Rendimiento	<ul style="list-style-type: none"> • Máxima altitud: 18.000 m • Máxima velocidad: 515 m/s • Aceleración: 4G
Puerto UART	<ul style="list-style-type: none"> • Velocidades de transmisión de 4800 bps a 115200 bps
Rango de temperaturas	<ul style="list-style-type: none"> • Operación normal: -40°C a 85°C

Fuente: Datasheet Quectel L80

3.2.1 Hardware Quectel L80

El diseño hardware del módulo Quectel L80 cuenta con 12 pines, que sirven para alimentación y comunicación del elemento; además posee una antena parche integrada. La Figura 17 detalla los pines del receptor GPS.

Figura 17. Pines del receptor GPS Quectel L80



Fuente: Datasheet Quectel L80

3.2.2 Protocolo de Comunicación para Quectel L80

El módulo Quectel L80 es compatible con el protocolo de comunicación NMEA 0183 comandos estándar. NMEA es el acrónimo de National Marine Electronics Association, que es un protocolo estándar de transmisión de datos para la comunicación, en un principio de la instrumentación marina. La información viene codificada en unos bloques de sentencias que el GPS transmite en código ASCII (American Standard Code) [19].

La especificación NMEA 0183 es el protocolo que transmite tramas GPS, los datos viajan en forma de "sentencias". Cada sentencia empieza con el símbolo "\$", seguido de dos caracteres que identifican al mensaje. Para tramas GPS, los caracteres son GP, después tres caracteres con la identificación del tipo de trama que se obtiene. Que pueden ser RMC (Recommend Minimum Specific GPS/TRANSIT Data), VTG (TrackMadeGood and GroundSpeed), GGA (Global PositioningSystemFix Data), GSA (GPS DOP and Active Satellites), GSV (GPS Satellites in View), GLL (Geographic Position, Latitude/Longitude), entre otras. [19] y [20]. A continuación se ejemplifica, algunas de las tramas que el receptor GPS otorga, separadas por comas y terminando con el símbolo "*" que indica el fin de los datos.

- Ejemplo trama GPGGA:
\$GPGGA,015540.000,3150.68378,N,11711.93139,E,1,08,0.6,0051.6,M,0.0,
M,,*58
La Tabla 7 explica la información de la trama GPGGA:

Tabla 7. Descripción de una trama GPGGA

Datos	Descripción
015540.000	Hora UTC, en formato hhmmss.sss: 01:55:40.000
3150.68378,N	Latitud en grados y minutos, en formato ddmm.mmmm: 31°50.68378` N representa norte S representa sur
11711.93139,E	Longitud en grados y minutos, en formato dddmm.mmmm: 117°11.93139` E representa este W representa oeste
1	Calidad de coordenadas 0 = nulas 1 = G.P.S 2 = DG.P.S
08	Número de satélites fijados, esta información tiene que estar aproximadamente entre 0 y 12.
0.6	Dilución de precisión horizontal.
0051.6,M	Altitud sobre el nivel del mar en metros.
Campo vacío	Tiempo desde la última actualización en segundos.
Campo vacío	Número de identificación de la estación DG.P.S.

Fuente: Datasheet Protocolo L80
URL: <http://www.quectel.com/es/product/prodetail.aspx?id=62>

- Ejemplo trama GPRMC:
\$GPRMC,013732.000,A,3150.7238,N,11711.7278,E,0.00,0.00,220413,,A*6
La Tabla 8 explica la información de la trama GPRMC:

Tabla 8. Descripción de una trama GPRMC

Datos	Descripción
013732.000	Hora UTC, en formato hhmmss.sss: 01:37:32.000
A	Validez de los datos V = Inválido A = Válido
3150.7238,N	Latitud en grados y minutos, en formato ddmm.mmmm: 31°50.7238` N representa norte S representa sur
11711.7278,E	Longitud en grados y minutos, en formato dddmm.mmmm: 117° 11.7278` E representa este W representa oeste
Campo vacío	Velocidad arriba del suelo en millas náuticas por hora.
Campo vacío	Dirección arriba del suelo en grados absolutos.
220413	Fecha en formato ddmmyy 22 de abril de 2013.

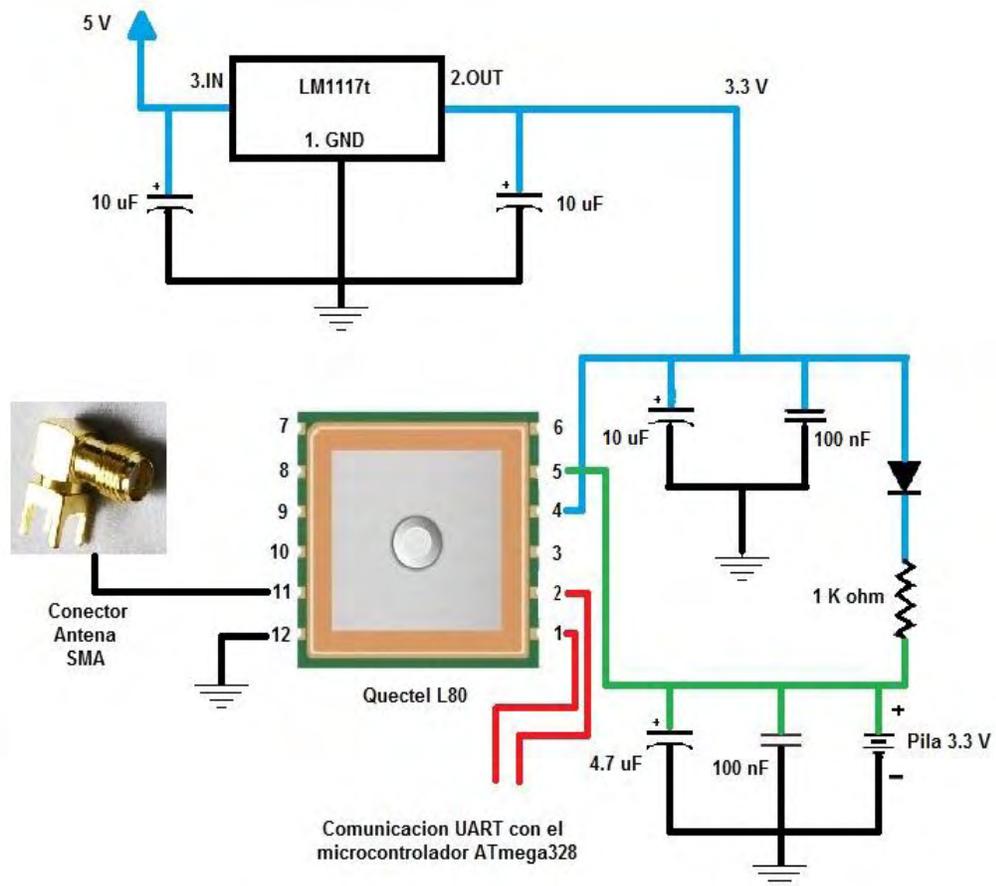
Fuente: Datasheet Protocolo L80

3.2.4 Esquema de Conexión del Receptor GPS Quectel L80

El módulo permite un montaje sencillo y eficaz. El voltaje de alimentación del Quectel L80 (**VCC**) se ha ajustado a 3.3V exactos por medio del regulador de voltaje LM1117t. El voltaje **V_BCKP** que sirve para que el módulo encuentre satélites lo más pronto posible a la hora de encenderlo, tiene un valor igual de 3.3V, pero este se otorga a través de una pila o batería.

El pin **1PPS** es una salida del módulo, que emite un pulso de 100 ms en el diseño realizado. Sin embargo, este pin no se lo tuvo en cuenta. El pin **TIMER** puede ser usado para el control de encendido del GPS, en caso de no usarlo se conecta a tierra. El **RESET** del elemento es activo bajo, además el pin **EX_ANT** es el que da la opción de conectarle al receptor una antena externa, con una impedancia de 50 ohm para mejorar la precisión del elemento. Finalmente, los pines 1 y 2 (**TXD1 Y RXD1**) son los que permiten la comunicación del elemento con el microcontrolador por medio de comunicación serial UART, la Figura 18 muestra el esquema de conexión utilizado en el dispositivo para el receptor GPS.

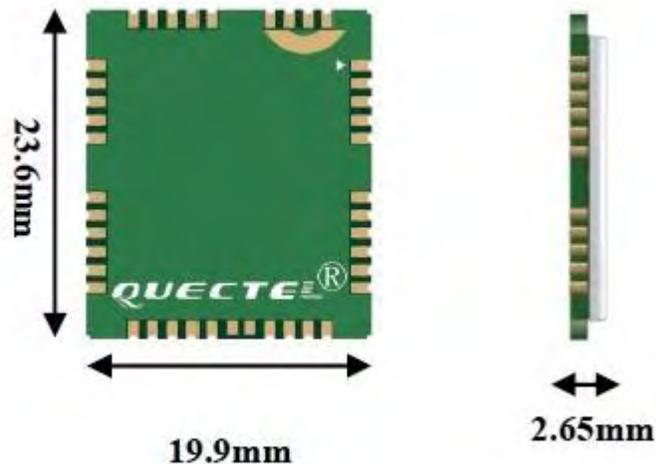
Figura 18. Esquema de conexión del receptor GPS Quectel L80



3.3 MÓDULO GSM/GPRS QUECTEL M95

El módulo GSM/GPRS Quectel M95 en el dispositivo ARGUS AUTOMOTIVE LM8095 es el encargado de transferir información desde cualquier lugar hacia un dispositivo móvil o un servidor. Este elemento es capaz de trabajar en distintas frecuencias (GSM850MHz, GSM900MHz, DCS1800MHz y PCS1900MHz), en protocolos de servicio de internet TCP/UDP (Transmission Control Protocol/User Datagram Protocol), FTP (File Transfer Protocol) y PPP,(Point-to-Point protocol) siendo accesible mediante comandos AT, y trabajando con un voltaje recomendado de 4.1V y un consumo de energía menor a 1 Ma. Además, a pesar de tener una buena cantidad de cualidades, posee un tamaño muy pequeño [21]. La Figura 19 muestra el módulo y sus dimensiones.

Figura 19. Dimensiones del módulo Quectel M95



Fuente: Datasheet Quectel M95

URL: http://www.quectel.com/UploadImage/Downlad/M95_Hardware_Design_V1.3.pdf

Por otro lado, el módulo Quectel M95, permite trabajar por medio de una interfaz SIM, y utiliza comunicación UART. En la Tabla 9 se muestra las principales características del componente mencionado.

Tabla 9. Principales características del módulo Quectel M95

Característica	Implementación
Fuente de alimentación	<ul style="list-style-type: none"> • Voltaje entre: 3.3V – 4.6V
Consumo de energía	<ul style="list-style-type: none"> • 0.9 mA
Bandas de frecuencia	<ul style="list-style-type: none"> • Cuatro bandas: GSM850, GSM900, DCS1800, PCS1900 • La banda de frecuencia puede ser escogida por comandos AT
Potencia de transmisión	<ul style="list-style-type: none"> • 2W para GSM850 y GSM900 • 1W para DCS1800 y PCS1900
Conectividad GPRS	<ul style="list-style-type: none"> • La conectividad es de clase B, esto quiere decir que no puede usarse GSM y GPRS al mismo tiempo
Rango de temperatura	<ul style="list-style-type: none"> • De -40°C hasta 85°C
Interfaz SIM	<ul style="list-style-type: none"> • Soporta SIM card de 1.8V hasta 3V
Interfaz UART	<ul style="list-style-type: none"> • Soporta velocidades de transmisión de 4800 bps a 115200 bps

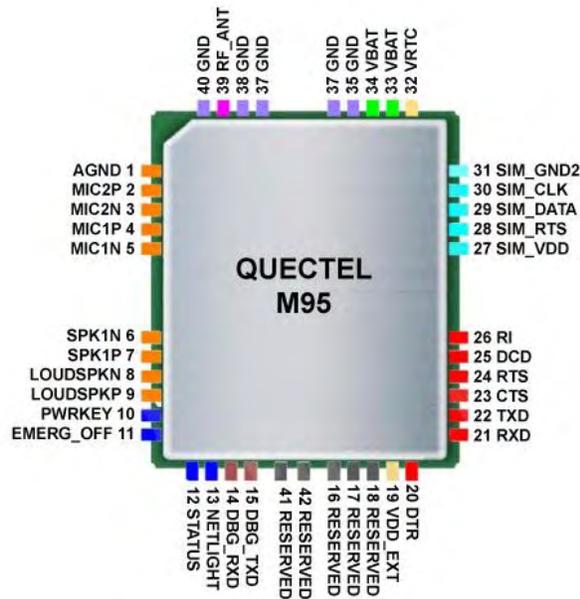
Fuente: Datasheet Quectel M95

Algo muy importante es que el módulo permite velocidades de transferencia máximas a 85.6 Kbps. Además, es un elemento ultraliviano con tan solo 3 gramos de peso, y requiere de una antena con impedancia de 50 Ohm para su óptimo funcionamiento.

3.3.1 Hardware Quectel M95

El hardware del módulo Quectel M95 cuenta con 42 pines, que sirven para la alimentación del elemento, la conexión de la interfaz de la SIM card, la comunicación UART del elemento, pines de encendido y apagado, pines de conexión de antena externa y pines para la interfaz de audio. La Figura 20 detalla los pines del módulo GPRS.

Figura 20. Pines del módulo GPRS Quectel M95



Fuente: ACME Systems

URL: http://www.acmesystems.it/terra_modem_gprs

El módulo presenta varias opciones para trabajar con él, pero en la presente investigación no se utiliza algunas interfaces que el M95 proporciona, como es el caso de la interfaz de audio y voz. Algunos de los pines principales y que son usados en el dispositivo ARGUS AUTOMOTIVE son los siguientes:

- El pin 10 (**PWRKEY**) que es usado para el encendido del componente. Con una señal de alto por aproximadamente 1 segundo. El pin 11 (**EMERG_OFF**) apaga el componente con una señal de mínimo 20 milisegundos.
- Pin 12 (**STATUS**) es salida de un pulso, usada para saber si el módulo está encendido.

- Pin 13 (**NETLIGHT**) es también una salida que indica el estado de red, dependiendo del tiempo de nivel alto de la señal y el tiempo de nivel bajo. La Tabla 10 muestra los estados de red.

Tabla 10. Estados de red del módulo Quectel M95

Estado	Función del módulo
off	El módulo no está corriendo
64 ms nivel alto / 800 ms nivel bajo	El módulo no está sincronizado con la red
64 ms nivel alto / 2000 ms nivel bajo	El módulo esta sincronizado con la red
64 ms nivel alto / 600 ms nivel bajo	La transferencia de datos GPRS está en curso

Fuente: Datasheet Quectel M95

- El pin 19 (**VDD_EXT**) es una salida que brinda 2.8V, para cualquier circuito externo que se desee montar.
- Los pines del 20 al 26, son utilizados para la comunicación UART del módulo.
- Los pines del 27 hasta el 31, hacen parte de la interfaz para el montaje de la SIM card.
- El módulo es alimentado en los pines 33 y 34 (**VBAT**) con un voltaje de aproximadamente 4.1V.
- En el pin 39 (**RF_ANT**) se debe conectar una antena externa de impedancia de 50 Ohm.

3.3.2 Protocolo de Comunicación para Quectel M95

El módulo GSM/GPRS Quectel M95 está integrado con protocolos de servicios de internet TCP/IP y comandos AT. En el manual del elemento Quectel M95 se presentan aproximadamente 180 comandos AT que pueden ser utilizados por el módulo [10], y que son los utilizados por ARGUS AUTOMOTIVE. Estos comandos se aplican al elemento por medio del puerto serial del microcontrolador ATmega328P-PU.

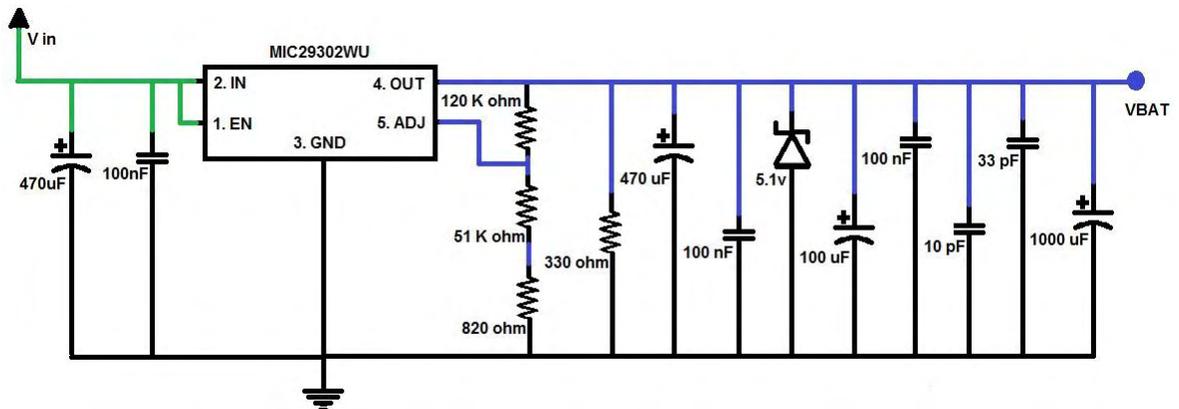
3.3.3 Esquema de Conexión del Módulo GSM/GPRS Quectel M95

Un objetivo específico de la investigación es el desarrollar un sistema de comunicación entre el dispositivo ubicado en el automóvil y el computador receptor de los datos de monitoreo y diagnóstico del mismo. Por esto se trabaja con comunicaciones móviles y específicamente con el módulo Quectel M95.

3.3.3.1 Etapa de Potencia para el Módulo Quectel M95

La etapa de potencia está compuesta por el regulador de voltaje ajustable MIC29302WU, que opera con hasta 3 A, soportando temperaturas de -40°C hasta 125°C . En esta investigación el regulador se ajustó para alimentar al módulo GPRS con 4.1V, el regulador tiene una etapa de filtros para limpiar algún tipo de ruido existente que pueda interferir en el óptimo funcionamiento del instrumento. La Figura 21 detalla el esquema de la etapa de potencia del M95.

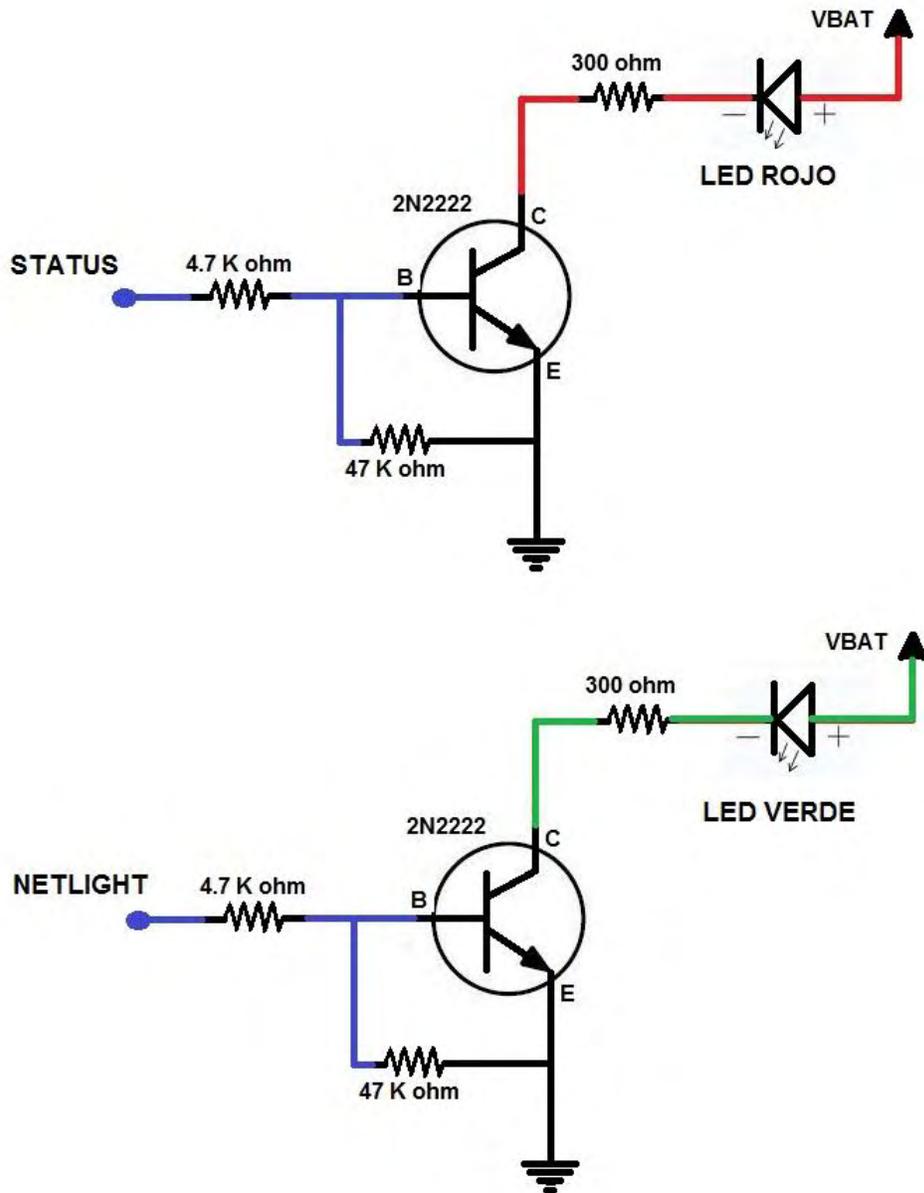
Figura 21. Etapa de potencia para el Módulo Quectel M95



3.3.3.2 Indicadores de STATUS y NETLIGHT

Los pines 12 y 13 son STATUS y NETLIGHT respectivamente, son salidas de pulsos que indican si el elemento esta encendido y el estado de red del mismo. Para visualizar esto en diodos LED, se adoptó un esquema elemental utilizando transistores 2N2222, tal como muestra la Figura 22.

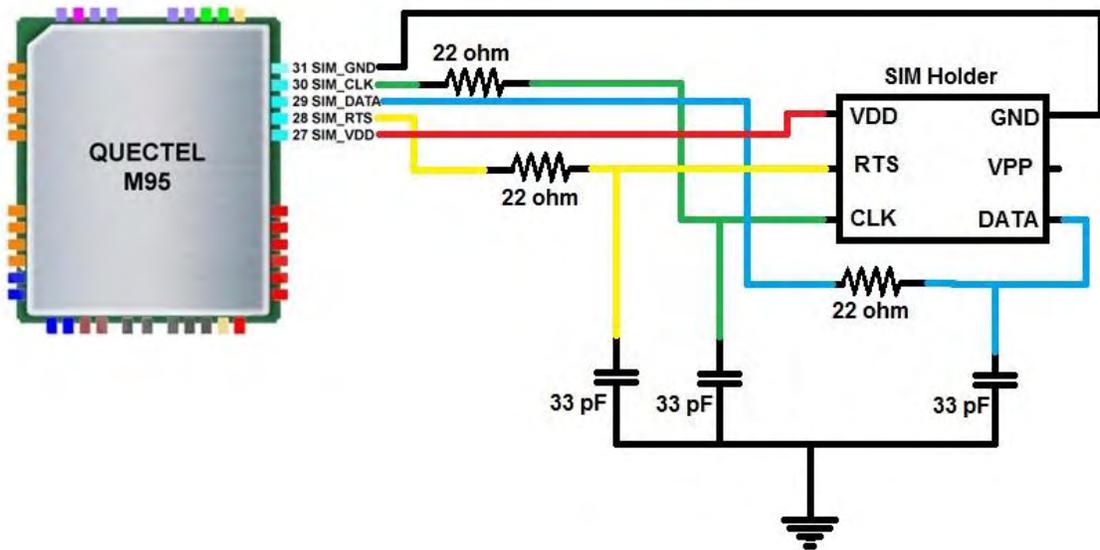
Figura 22. Esquema de indicadores de los pines STATUS y NETLIGHT



3.3.3.3 Interfaz SIM Card del Módulo Quectel M95

El esquema para el funcionamiento de la SIM card es sencillo. Lo más destacado en este ítem, es que el módulo M95 sólo soporta trabajar con SIM card que funcionen de 1.8V hasta 3V. La Figura 23 detalla el montaje para poder operar con una tarjeta SIM en el módulo GPRS.

Figura 23. Esquema para la interfaz SIM card



3.3.3.4 Interfaz de Comunicación UART para el Quectel M95

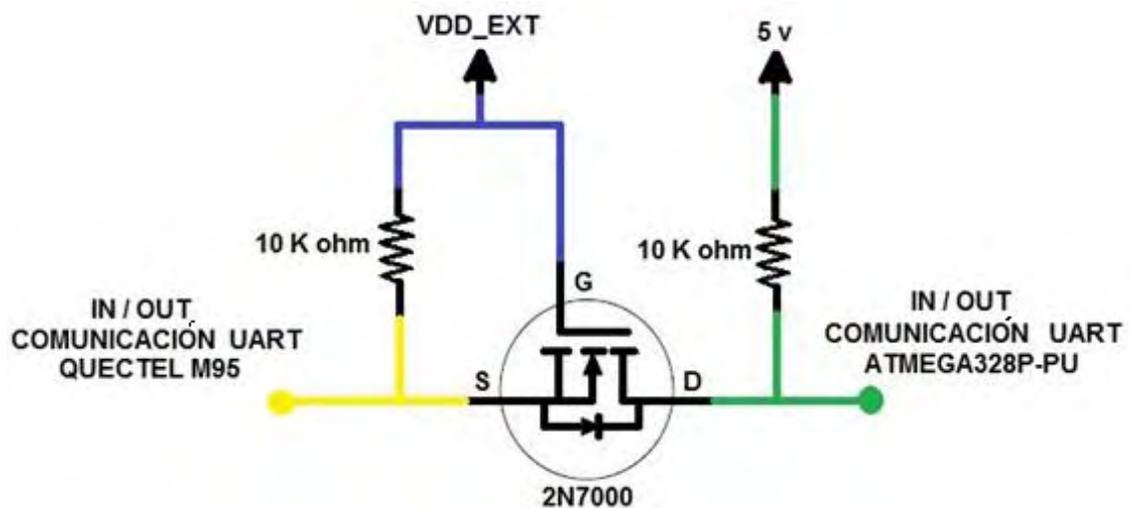
La interfaz para la comunicación del elemento Quectel M95 por medio de puerto UART, se encuentra desde el pin 20 hasta el 26, entre los que se localiza la línea de transferencia de datos (**TX**) y la línea de recepción de datos (**RX**), que son pines fundamentales para establecer una comunicación con otro dispositivo. En este caso, la comunicación se tiene que establecer con el elemento ATmega328P-PU.

Como se mencionó anteriormente, el elemento Quectel M95 opera con un voltaje de 4.1V, y la línea de comunicación **RX** del mismo soporta un máximo de 3.3V, mientras que el circuito ATmega328P-PU opera con un voltaje de 5V, por lo que se debe implementar una fase de acople para que el sistema GPRS/GSM trabaje sin problemas y sin poner en riesgo su funcionalidad. Para poder adaptar los diferentes niveles de voltaje de los elementos se realiza un arreglo de transistores MOSFET, ya que las velocidades de transmisión son muy altas, y el MOSFET de referencia 2N7000 puede operarlas sin problema alguno.

3.3.3.4.1 Etapa de Acople de Voltaje por Medio de Transistores 2N7000

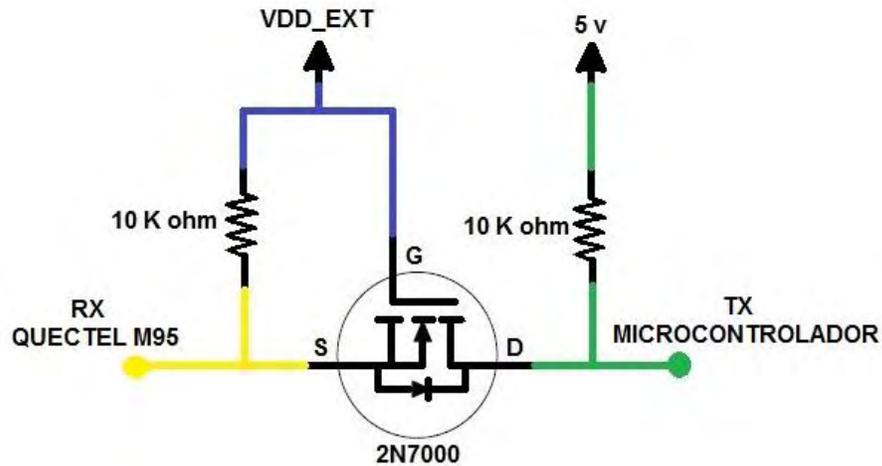
El esquema usado es muy funcional porque es un modelo bidireccional. En la Figura 24 se especifica la forma en la que el transistor es conectado para permitir la comunicación entre el ATmega328P-PU y el módulo Quectel M95. El circuito necesita de dos voltajes de alimentación, que son de los dos elementos que se requiere comunicar, o sea **VDD_EXT** que es aproximadamente 3.3V perteneciente al voltaje de comunicación del M95 y los 5V que son el voltaje de alimentación del microcontrolador.

Figura 24. Esquema para el acople de voltajes



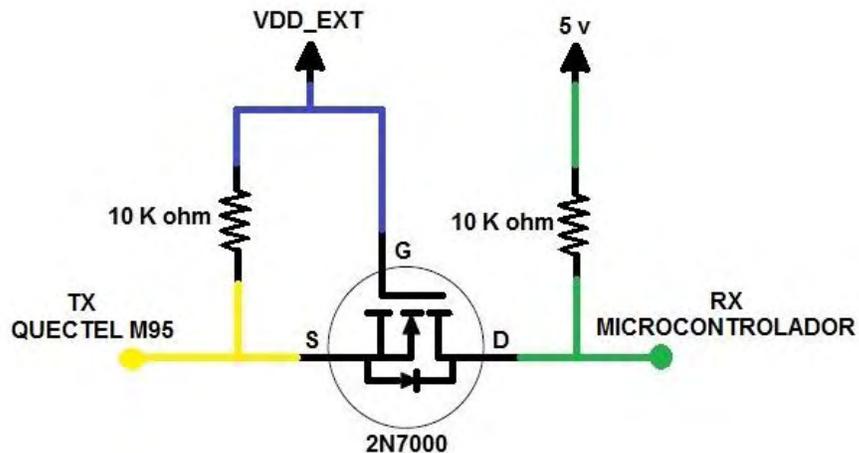
La Figura 25 hace referencia a la forma en la que se debe conectar el pin 21 (**RX**), para que se pueda acoplar al voltaje que maneja el microcontrolador. En este caso si el ATmega328P-PU transmite un cero (0) lógico, entonces, se polariza el diodo interno del transistor, haciendo que su voltaje sea prácticamente cero y el voltaje entre G y S (V_{gs}) será lo suficientemente grande para polarizar el transistor. Esto baja su resistencia interna logrando que el cero que se transmite llegue a su destino con otro nivel de voltaje. Si por lo contrario el microcontrolador transmite un uno (1) lógico, lo que sucede es mucho más sencillo ya que el transistor no se polariza, no hay conexión entre circuitos y entonces en el Quectel M95 se recibirá también un uno (1), pero con menos nivel de voltaje debido a los 3.3V que otorga VDD_EXT [22].

Figura 25. Acople entre RX del Quectel M95 y el TX del microcontrolador



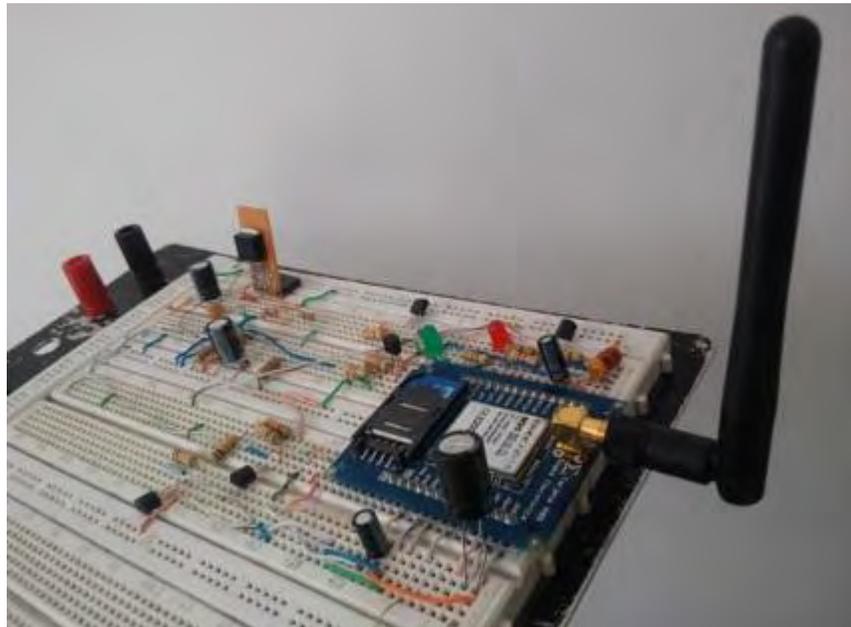
La Figura 26, muestra como se debe conectar el pin 22 (TX), para acoplar su voltaje con el del microcontrolador. Ahora si el módulo Quectel M95 es el que transmite un cero (0) lógico, entonces el voltaje entre G y S (V_{gs}) es V_{DD_EXT} (3.3V), de este modo el transistor se polariza y la resistencia del transistor se vuelve casi cero, por lo que el cero (0) lógico pasa al otro lado, y el valor que le llega al microcontrolador es el esperado cero (0), y por lo contrario, si el módulo Quectel M95 transmite un uno (1) lógico, entonces el voltaje entre G y S (V_{gs}) es cero, por lo que el transistor no se polariza, haciendo que la resistencia entre tierra y D sea muy alta, aislando ambos circuitos, y haciendo que al microcontrolador le llegue un uno (1), correspondiendo a los 5V de la alimentación [22].

Figura 26. Acople entre TX del Quectel M95 y el RX del microcontrolador



Finalmente, la Figura 27 muestra el primer montaje del módulo GSM/GPRS Quectel M95 realizado en protoboard.

Figura 27. Circuito del módulo Quectel M95 en protoboard



3.4 DESARROLLO DEL SISTEMA CAN BUS

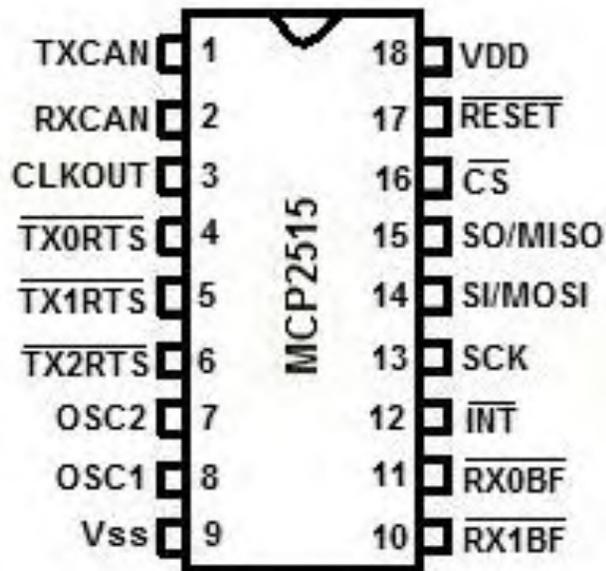
Para la adquisición de los datos más importantes a monitorear en el vehículo, se construyó un prototipo de escudo para CAN Bus, que adopta un controlador CAN MCP2515 que trabaja con interfaz periférica serie SPI. Este módulo es capaz de transmitir y recibir datos tanto estándar como extendidos, posee máscaras y filtros de aceptación que se utilizan para filtrar los mensajes no deseados. Además, el dispositivo se basa en un transceptor MCP2551 que sirve como interfaz entre el bus físico y el controlador CAN.

3.4.1 Controlador CAN MCP2515

El circuito integrado MCP2515 es un controlador de segunda generación y autónomo para aplicaciones relacionadas con buses CAN y tiene la capacidad de comunicarse con el microcontrolador y todos los dispositivos del vehículo por medio de la interfaz SPI, para aplicaciones de alta velocidad, el controlador implementa la especificaciones de CAN 2.0B capaz de transmitir y de enviar a la vez tramas estándar y tramas extendidas con una velocidad de hasta 1Mb/s.

El diseño de hardware del controlador CAN MCP2515 cuenta con 18 pines, algunos de estos sirven para la alimentación, otros para la comunicación con el microcontrolador y sus respectivos buffers de transmisión y recepción, entre otros. En la Figura 28 se detallan los pines del MCP2515.

Figura 28. Diagrama de pines del controlador CAN MCP2515



En la Tabla 11 se puede observar algunos de los parámetros del controlador que se debe tener en cuenta para su correcto funcionamiento:

Tabla 11. Parámetros del controlador MCP2515

Parámetro	Valor
Versión CAN soportada	CAN 2.0B
TX Buffers	3
RX Buffers	2
Máscaras	2
Interruptor de salida	1
Filtros	6
Rango de temperatura (°c)	-40 a 125
Voltaje de operación (v)	2.5 a 5.5

Fuente: Datasheet MCP2515

URL: <http://www.microchip.com/wwwproducts/en/MCP2515>

Para la transmisión de mensajes, el MCP2515 implementa tres buffers de transmisión, donde para cada mensaje a transmitir, primero se hace un almacenamiento intermedio con el fin de determinar las condiciones bajo las cuales el mensaje se transmitirá. Posteriormente se analiza qué mensaje tiene mayor prioridad para ser transmitido, prioridad que se determina de acuerdo a la configuración que se haya hecho anteriormente al controlador con respecto al grado de prioridad de cada uno de sus buffers.

El controlador tiene incluido para la recepción de mensajes dos buffers, además cuenta con filtros y máscaras de aceptación para determinar si el mensaje es guardado o no en la memoria. Para la programación, el controlador CAN tiene los siguientes modos de operación:

- Modo configuración (Configuration Mode): el MCP2515 debe ser inicializado antes de la activación, que sólo es posible si el dispositivo está en modo configuración. Este modo se selecciona automáticamente después del encendido del dispositivo, y es aquí donde se pueden modificar algunos parámetros como por ejemplo el registro de filtros y máscaras. Además si en el dispositivo se presenta algún error, al estar seleccionado este modo hará que todo error se borre.
- Modo normal (Normal Mode): es el modo de funcionamiento estándar del MCP2515. En este modo el dispositivo monitorea activamente todos los mensajes presentes en el bus, además tiene la capacidad de generar tramas de error, y es el único modo en el que el MCP2515 transmitirá mensajes a través del CAN Bus, característica que es muy importante.
- Modo sueño (Sleep Mode): Este modo se lo utiliza para minimizar el consumo de corriente del dispositivo. La interfaz SPI seguirá funcionando normalmente para que se tengan acceso a todos los registros, además al estar en este modo el MCP2515 detiene su oscilador interno. Posteriormente se pondrá en marcha cuando se produzca una actividad del bus o cuando lo establezca la computadora central.
- Modo solo escucha (Listen-Only Mode): Este modo proporciona al controlador únicamente un medio para recibir los mensajes, incluidos los mensajes que contengan algún tipo de error. Es muy utilizado para desarrollar aplicaciones de monitoreo de bus como es el caso del dispositivo que se desarrolla en este trabajo, el cual siempre está recibiendo los mensajes de las variables de monitoreo, además de los mensajes de diagnóstico, permaneciendo en un constante monitoreo al bus.
- Modo retorno (Loopback Mode): Este modo permite que se pueda realizar la transmisión interna de mensajes desde los buffers de transmisión hasta los buffers de recepción sin llegar a la transmisión de mensajes al CAN Bus. En esta configuración las máscaras se pueden ajustar a cero para poder aceptar todos los mensajes del bus.

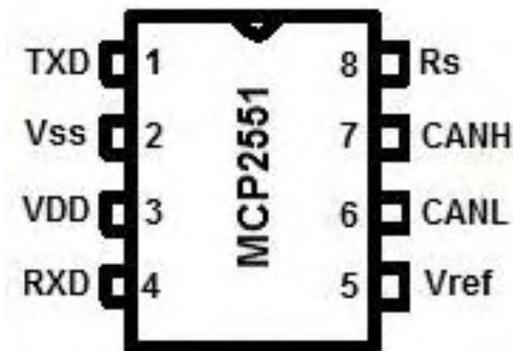
3.4.2 Transceptor MCP2551

Es un transceptor de alta velocidad que sirve de interfaz entre un controlador de protocolo CAN, para éste caso el MCP2515, y el bus físico. Este transceptor proporciona una transmisión diferencial y tiene la capacidad de recibir cualquier tipo de tramas del protocolo CAN, es totalmente compatible con la norma ISO 11898, y es capaz de funcionar a velocidades de hasta 1Mbits/s. Es necesario que cada nodo conectado a una red CAN tenga un dispositivo para convertir las señales digitales generadas por un controlador CAN a las señales adecuadas para la transmisión a través del cable del bus. Las salidas del MCP2551 impulsarán una carga mínima de 45 ohmios.

Una de las características del transceptor es la protección interna con la que cuenta ante una descarga de corriente excesiva, por el método de apagado térmico del circuito que desactiva los controladores de salida cuando la temperatura excede el valor límite nominal de 165°C. Las otras partes del chip seguirán operando normalmente y pasado unos minutos la temperatura del transceptor bajará debido a la disminución de la disipación de potencia en las salidas del transmisor.

Para la elaboración de la tarjeta se utilizó el transceptor en un encapsulado PDIP (Plastic Dual In-Line Package) que cuenta con 8 pines. En la Figura 29 se puede observar el hardware del transceptor MCP2551 con el nombre de cada uno de sus pines.

Figura 29. Diagrama de pines de transceptor MCP2551



En la Tabla 12 se puede observar la descripción de los pines que tiene el transceptor MCP2551 con sus respectivos nombres y la función de cada uno.

Tabla 12. Hardware MCP2551

Número Pin	Nombre Pin	Función Pin
1	TXD	Entrada de datos a transmitir
2	VSS	Tierra
3	VDD	Voltaje
4	RXD	Salida de datos de recepción
5	VREF	Referencia salida de voltaje
6	CANL	Línea CAN Nivel-Bajo de Voltaje
7	CANH	Línea CAN Nivel-Alto de Voltaje
8	RS	Entrada Control

Fuente: Datasheet MCP2551

URL: <http://www.alldatasheet.com/view.jsp?Searchword=Mcp2551>

El pin **RS** (pin 8) permite elegir entre tres modos de funcionamiento del transceptor:

- Modo High-Speed: este modo se selecciona conectando el pin RS a VSS. Aquí los controladores de salida de transmisión tienen una respuesta rápida ante una subida o caída de las líneas del bus.
- Modo Slope-control: esta configuración reduce enormemente las emisiones electromagnéticas, disminuyendo los tiempos de subida y caída de CANH y CANL. Este control se logra mediante la conexión de una resistencia entre RS y Ground (tierra). Aquí la tasa de ralentización es proporcional a la

salida actual en RS; debido a que la corriente está determinada por el valor de la resistencia que se tenga conectada.

- Modo Standby: el dispositivo se puede colocar en modo standby o sleep mediante la aplicación de un alto nivel en el pin RS. En modo Sleep el transistor se apaga y el receptor seguirá funcionando pero con una corriente muy baja, es decir el pin **RXD** seguirá operando pero de forma más lenta.

Cuando el transceptor está en modo High-Speed o Slope-control, los controladores para las señales CANH y CANL están regulados internamente para proporcionar simetría controlada con el fin de minimizar las emisiones electromagnéticas, para evitar que el transceptor se recaliente.

También es importante mencionar que si el MCP2551 detecta un estado bajo muy prolongado en la entrada del pin **TXD**, se desactivarán los controladores de las salidas de CANH y CANL con el fin de proteger los datos del CAN Bus. Los controladores se desactivarán si **TXD** está en estado bajo por más de 1,25 milisegundos como mínimo. Cuando se encuentre un flanco de subida en **TXD**, se restablecerá la lógica del temporizador y los controladores de salida de CANH y CANL volverá a retomar sus funciones.

3.4.3 Esquema de Conexión CAN Bus

Para el diseño del sistema de adquisición de datos por medio de la red CAN Bus, se tuvo en cuenta algunos dispositivos existentes en el mercado, como lo son: CAN-Bus Shield de Sparkfun y CAN-Bus Shield de Seed Studio. Tienen como principal base la conexión del controlador CAN MCP2515 y el transceptor MCP2551.

3.4.3.1 Conexión del Controlador MCP2515

Para adquirir datos de la red CAN Bus con el MCP2515, se ajustó un voltaje de alimentación de 5V, conectándolo a sus pines **VSS** y **VDD** (pin 9 y 18), los pines **TXCAN** y **RXCAN** (Pin 1 y 2) se conectan al receptor y transmisor (RX y TX) respectivamente del microcontrolador ATmega328P-PU. Los pines **TX0RTS**, **TX1RTS** y **TX2RTS** (pin 4,5 y 6) son los tres buffers que tiene el controlador para la transmisión, ocupando cada uno de estos 14 bytes de la memoria. El pin **RX0BF** (pin 11) es uno de los buffers de recepción y el de mayor prioridad cuando llegue algún mensaje, que cuenta con una máscara y dos filtros de aceptación de mensaje.

Otro de los buffers de recepción del controlador es el pin **RX1BF** (pin 10), que tiene una prioridad más baja, y cuenta con una máscara y cuatro filtros de aceptación de mensajes. En los pines **OSC1** y **OSC2** (pin 7 y 8) se conecta un

oscilador de 16MHz tal como lo indica el datasheet del componente, acompañado de unos condensadores cerámicos que tienen un valor de 22pf. El pin **CLKOUT/SOF** (pin 3) se puede utilizar como el reloj principal del sistema o como una entrada de reloj para otros dispositivos del sistema.

Este controlador posee un pin para su restablecimiento (**RESET** pin 17) que se debe colocar en bajo, asegurando así que la lógica y los registros vuelvan a su estado predeterminado. EL MCP2515 tiene ocho tipos de interrupciones, cuando se produce una interrupción el pin **INT** (pin 12) se coloca en estado bajo, y permanecerá así hasta que se despeje la interrupción. Las interrupciones pueden ser generadas por ejemplo, para indicar que un mensaje válido ha sido recibido o también cuando algún mensaje comenzó a transmitirse.

Los pines **SCK**, **SI**, **SO** y **CS** (pin 13, 14, 15 y 16) respectivamente, se utilizan para la comunicación SPI con el microcontrolador. Estos incluyen la línea de reloj (SCK) donde por cada pulso que se genere se podrá leer o enviar un bit, la línea de salida y entrada de datos (SI y SO) y por último un pin para activar o desactivar el controlador (CS), que permanece en bajo cuando el MCP2515 no está realizando ninguna operación.

3.4.3.2 Conexión del Transceptor MCP2551

Para la alimentación del MCP2551 se ajusta un voltaje de 5V y se conecta a los pines **VSS** y **VDD** (pin 2 y pin 3), el pin **TXD** y **RXD** (pin 1 y pin 4) son los pines de transmisión y recepción de datos que están conectados a los pines TXCAN y RXCAN del MCP2515 respectivamente. Los pines **CANL** y **CANH** (pin 6 y pin 7) se conectan con las dos líneas del CAN Bus a través de un conector de 9 pines ubicado en la tarjeta y el conector OBD-II del automóvil. Por último el pin **RS** (pin 8) se conecta a tierra y en medio de estos se coloca una resistencia.

En la Figura 30 se muestra de forma explicativa la conexión del controlador MCP2515 para que trabaje como una red CAN Bus e interactúe por medio del transceptor MCP2551 con la red CAN Bus del automóvil. Y en la Figura 31 se especifica la conexión tal como se plantea en el esquema final del proyecto utilizando el sistema OBD-II del automóvil, y detallando la conexión con el microcontrolador ATmega328P-PU, con el cual trabaja vía comunicación SPI.

Figura 30. Interacción entre red CAN Bus del automóvil y módulo CAN Bus

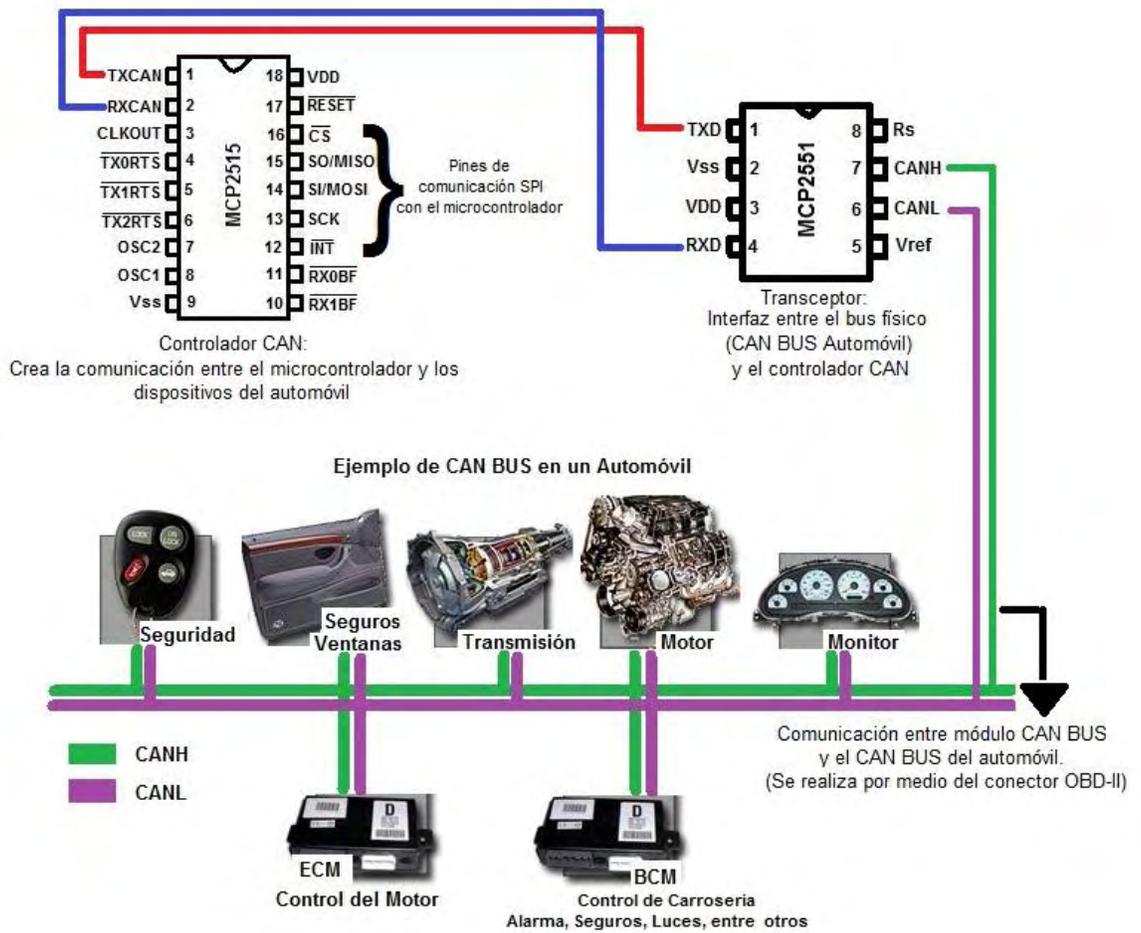
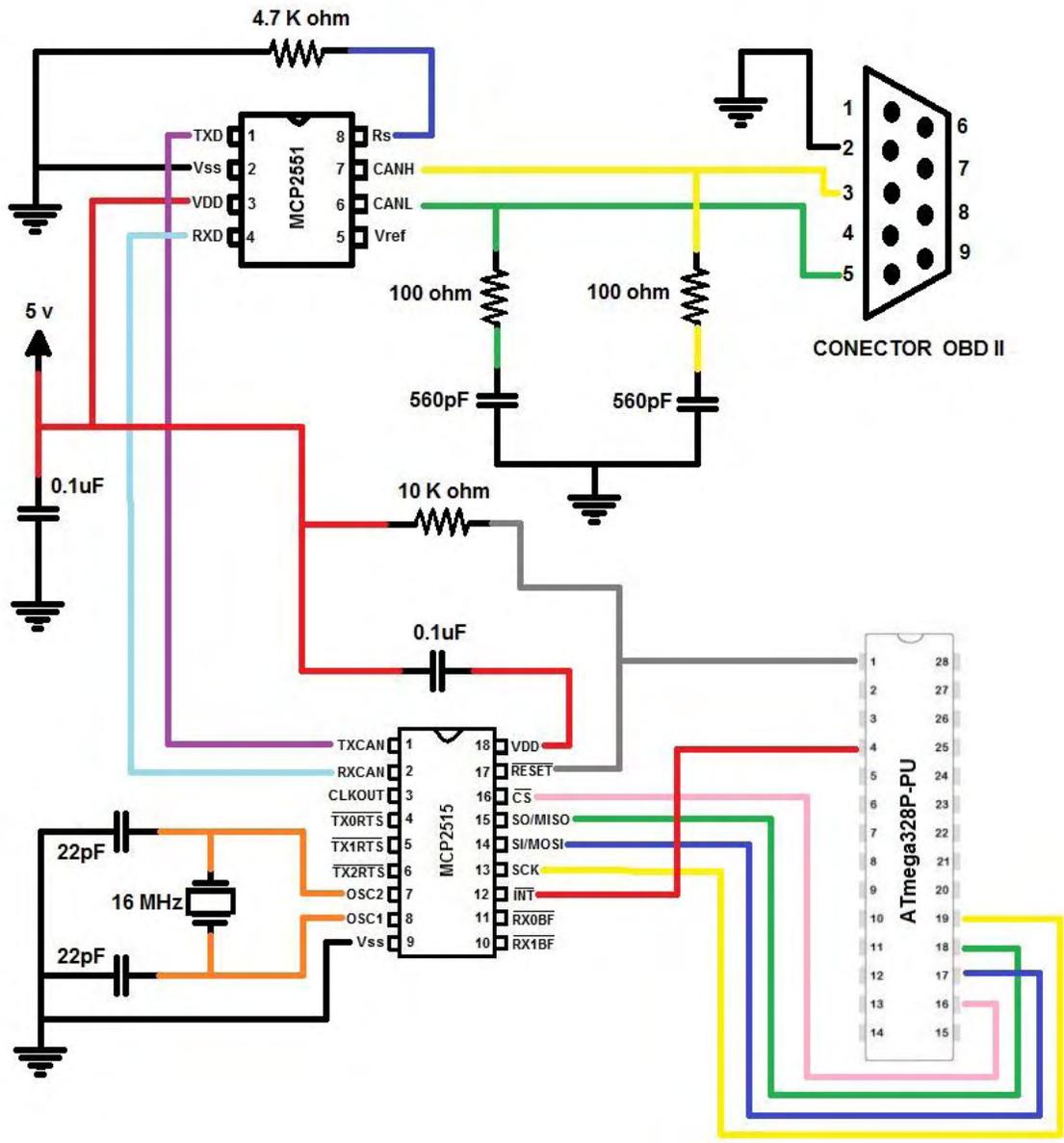
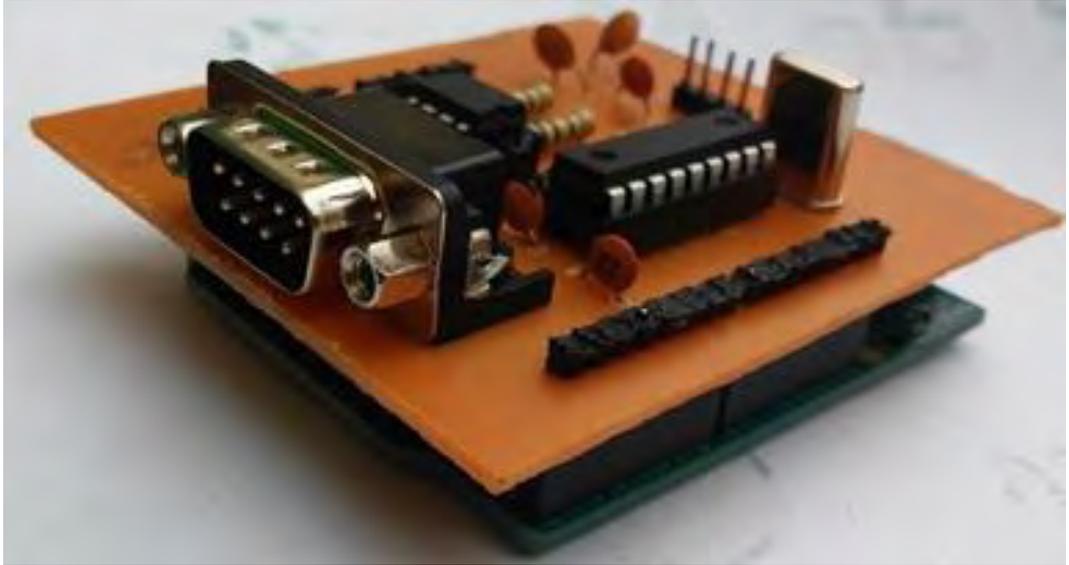


Figura 31. Esquema de conexión CAN Bus



En el esquema anterior se observa el conector de 9 pines OBD II, al cual le llegan las líneas CAN_H y CAN_L, y la línea de tierra, líneas que se conectan directamente al automóvil por medio de un cable de diagnóstico. En la Figura 32 se ilustra la primer fabricación del sistema CAN Bus.

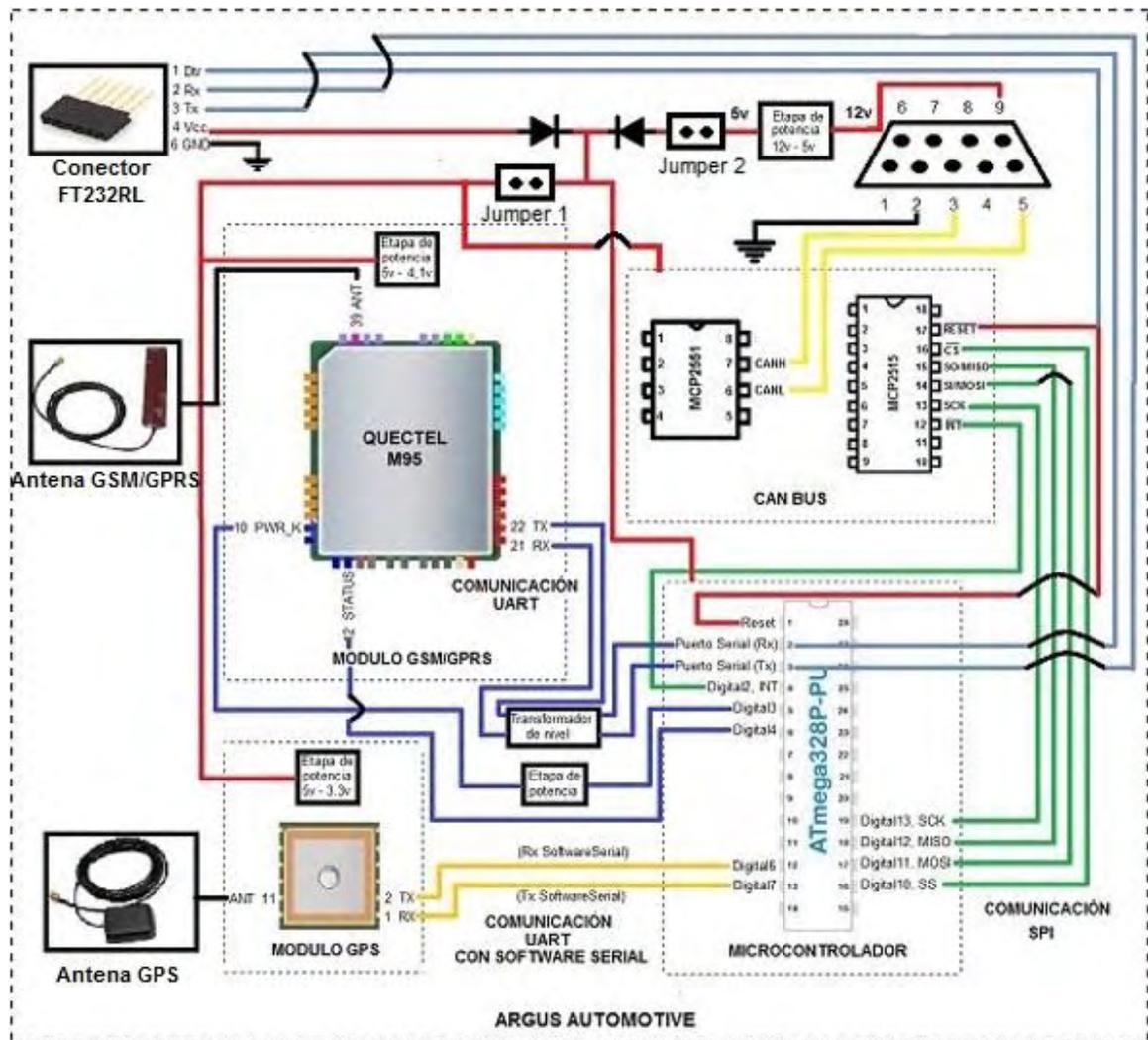
Figura 32. Primera fabricación del sistema CAN Bus



4. ESQUEMA Y FABRICACIÓN DEL DISPOSITIVO FINAL

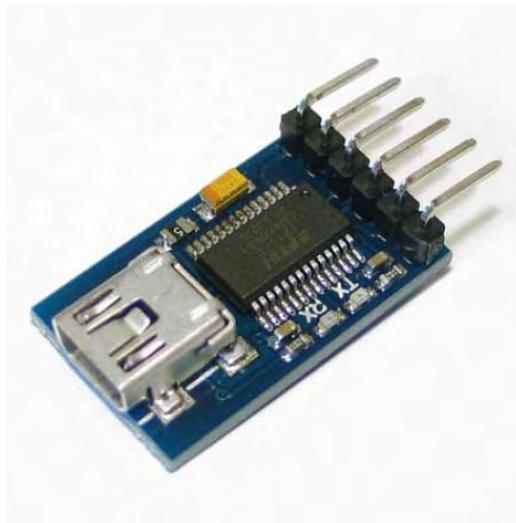
Anteriormente se han representado los esquemas detallados de cada uno de los elementos que componen el diseño de ARGUS AUTOMOTIVE. La Figura 33 muestra un circuito general de todos los anteriores esquemas unidos, el cual es el montaje que finalmente se toma en cuenta para la producción de la tarjeta. En ella se puede apreciar que el conector OBD-II del automóvil, es el encargado de alimentar a ARGUS AUTOMOTIVE cuando esta no está siendo programada.

Figura 33. Esquema general ARGUS AUTOMOTIVE



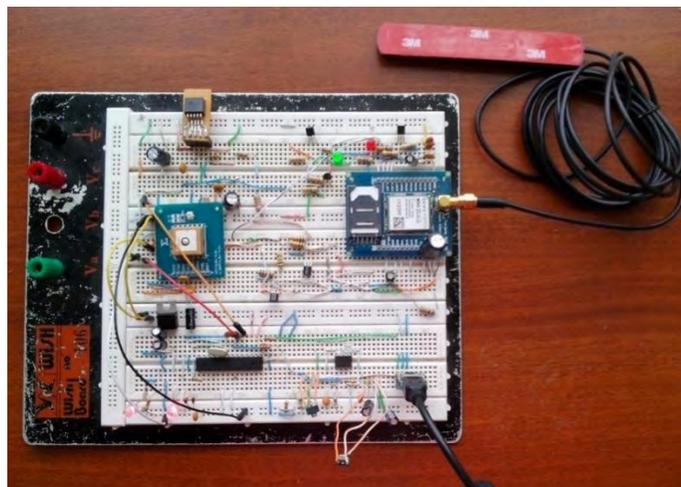
También, en la Figura 33 se observan pines de conexión tipo hembra (Dtr,Rx,Tx,Vcc,Cts,Gnd), por los cuales se programa la tarjeta, esto se debe algunos parámetros de diseño. La programación se la hace a través del integrado FT232RL, el cual transforma comunicación UART a USB. El elemento se puede apreciar en la Figura 34.

Figura 34. Conector FT232RL



Posteriormente, al tener en claro el esquema del diseño, se procedió a montar el proyecto en una protoboard como se ve en la Figura 35, lo cual sirvió para perfeccionar parámetros, y descartar fallas.

Figura 35. Montaje en protoboard de ARGUS AUTOMOTIVE



Una vez ensayado el esquema en protoboard y ver que su funcionamiento era apropiado, se procede a realizar el diseño en una plaqueta de circuito impreso PCB (Printed Circuit Board). Por medio de método casero de planchado, primero se enrutó pistas de conexión gracias al programa EAGLE. El resultante fue un circuito de aproximadamente 20cm de largo con 15cm de ancho. Estas rutas se

las imprime a láser y después se la plancha durante aproximadamente 30 minutos sobre una placa de cobre. La Figura 36 muestra el resultado de este proceso.

Figura 36. Esquema circuito impreso en placa PCB



Posteriormente, como se muestra en la Figura 37, se sumerge la placa de cobre en ácido férrico, lo que consume todo el cobre de la placa con excepción del que está cubierto por las rutas de tinta láser. La Figura 38 indica como se ve la placa después de ser sumergida en ácido.

Figura 37. Placa PCB en ácido férrico

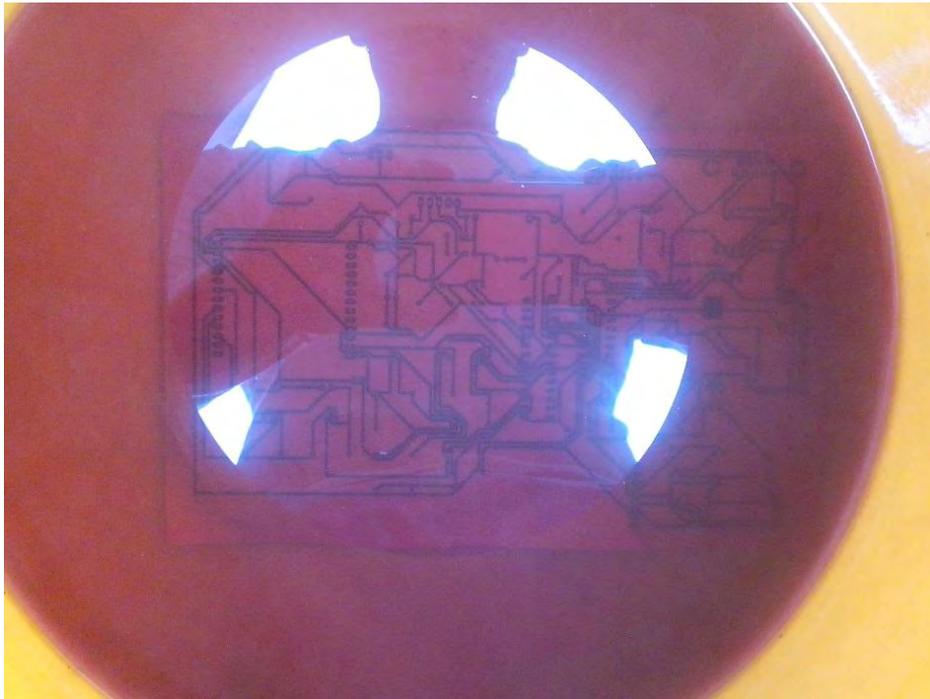
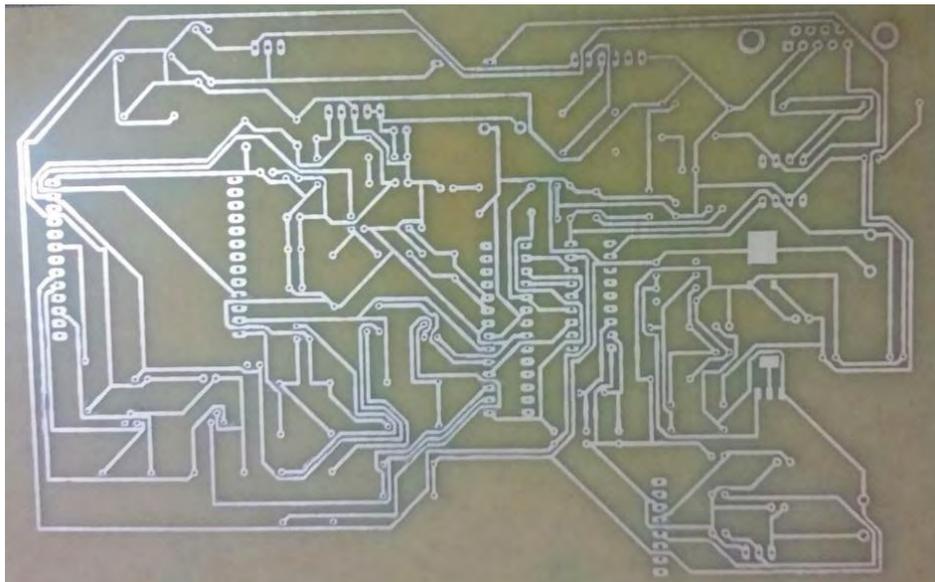
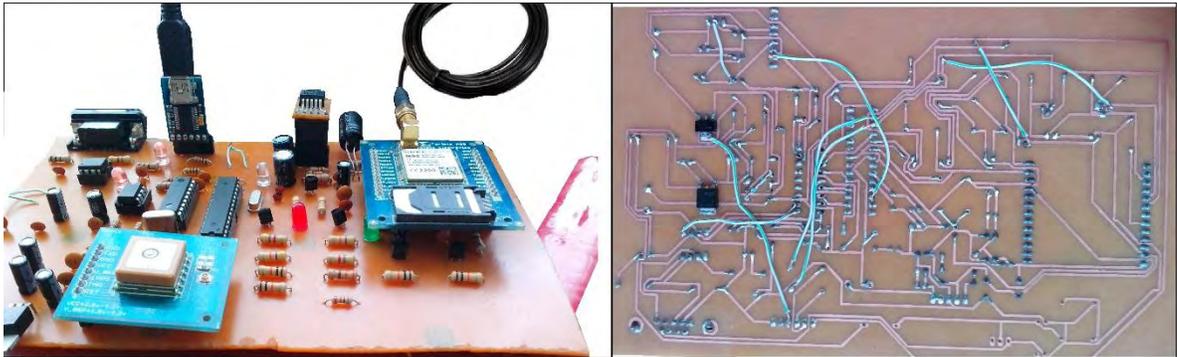


Figura 38. Pistas en cobre del esquema ARGUS AUTOMOTIVE



El paso a seguir fue taladrar la tarjeta, para introducir todos los elementos y módulos del diseño. Algunas pistas no las pudo enrutar el programa EAGLE, razón por la cual se soldaron algunos cables para completar las conexiones del esquema original. La Figura 39 muestra la primer fabricación del ARGUS AUTOMOTIVE.

Figura 39. Primer prototipo tarjeta ARGUS AUTOMOTIVE



El circuito resultante es funcional, pero su construcción no es de tipo profesional, para su presentación como prototipo se desea que el proyecto tenga alta calidad, por tal razón se estableció comunicación con COLCIRCUITOS manufactura electrónica digital, una empresa ubicada en la ciudad de Medellín dedicada a la fabricación de PCBs de alta complejidad. Con el diseño en EAGLE y todos los archivos que derivan de este programa, y las rutas diseñadas en doble capa para el proyecto (Figura 40), la empresa se encargó de la fabricación de la placa. Por un costo adicional realizaron el ensamblaje de los componentes más reducidos. Cabe resaltar que los componentes de la tarjeta fueron comprados por aparte, y enviados a COLCIRCUITOS. La Figura 41 muestra la cotización de la tarjeta realizada.

Figura 40. Rutas de ARGUS AUTOMOTIVE en EAGLE

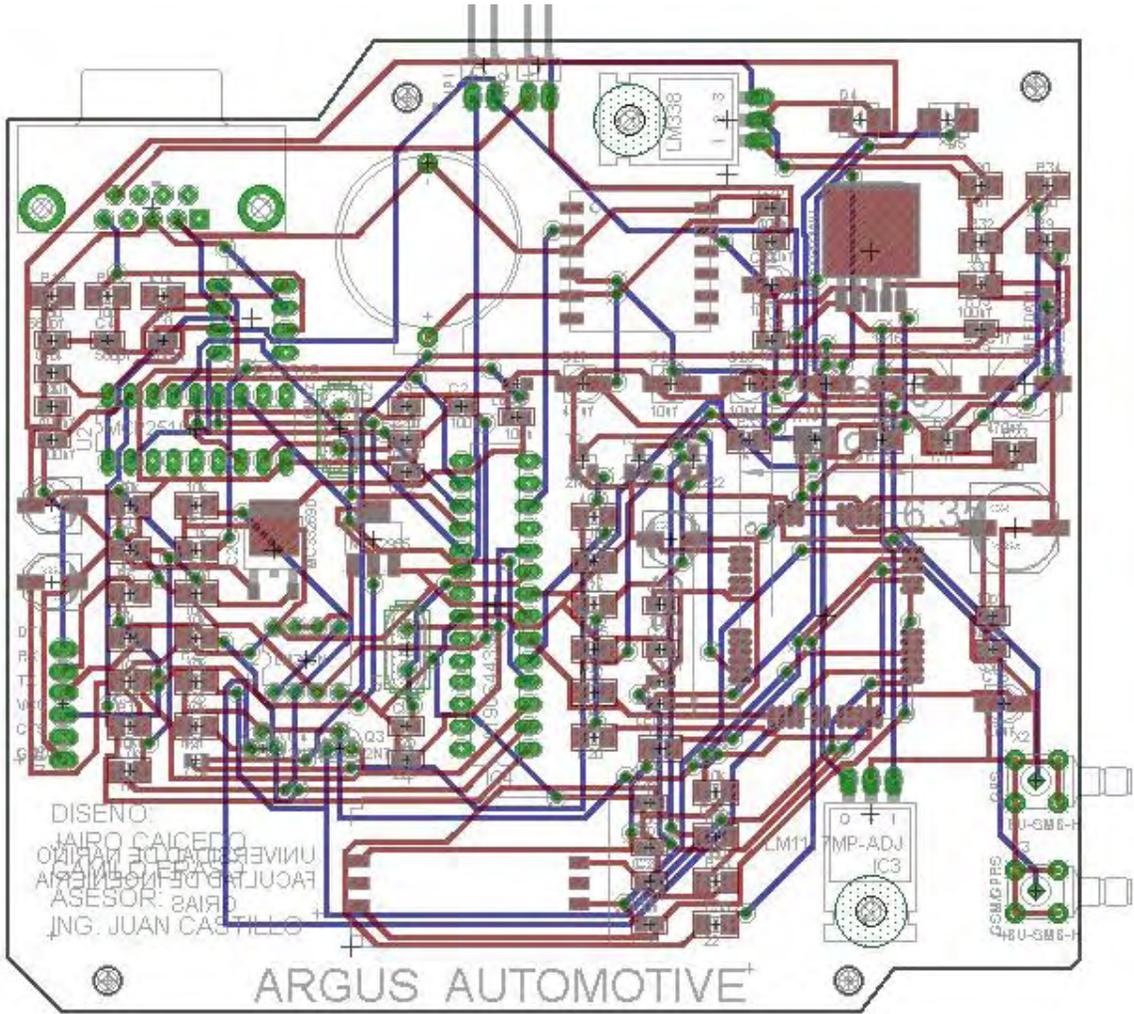


Figura 41. Cotización para el ensamble de ARGUS AUTOMOTIVE



Medellín, 14 de diciembre de 2015

Señor(a)
JAIRO ADRIAN CAICEDO BRAVO
 PASTO

Cordial saludo:

La presente es la cotización para el ensamble del (los) proyecto (s) solicitado (s) por usted (es):

NOMBRE DEL PROYECTO: **ESQUEMA**

CANTIDAD	CALIDAD DEL CIRCUITO	PROCESO	VALOR UNITARIO POR PROCESO*	VALOR UNITARIO TOTAL	VALOR TOTAL	VALOR DEL HERRAMENTAL*	FLETE	TOTAL + IVA
1	FORMATO ESTANDAR	CIRCUITO	\$72.000	\$133.238	\$133.238	\$220.764	\$ 26.952	\$437.594
		ENSAMBLE	\$61.238					
3	FORMATO ESTANDAR	CIRCUITO	\$39.050	\$72.939	\$218.816	\$220.764	\$ 27.808	\$537.721
		ENSAMBLE	\$33.889					

El prototipo final de ARGUS AUTOMOTIVE se fabricó en las instalaciones de COLCIRCUITOS, aproximadamente en un mes. La Figura 42 muestra el resultado final del proyecto diseñado.

Figura 42. Dispositivo final



Los componentes utilizados en este proyecto no son de fácil adquisición y ningún módulo ni circuito integrado pudo ser encontrado en la ciudad Pasto. Por esto los elementos fueron adquiridos en varias ciudades de Colombia, a excepción de los integrados MCP2515 y MCP2551 utilizados para la adquisición de datos del carro. Estos componentes no pudieron ser encontrados en Colombia, por tal razón se pidió la ayuda de Electrónicos Caldas, para que los importara desde Estados Unidos, con el triple del valor normal y con un tiempo de entrega aproximado de 2 meses. La Tabla 13 muestra el valor total aproximado de la tarjeta ARGUS AUTOMOTIVE.

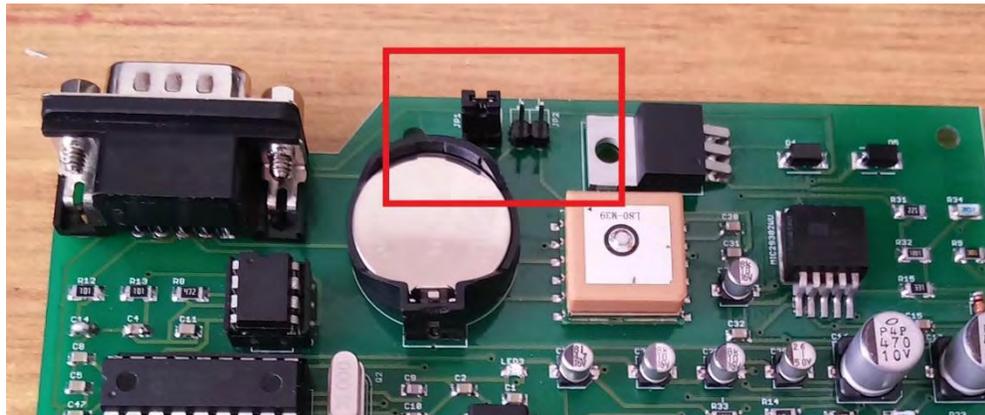
Tabla 13. Valor aproximado tarjeta ARGUS AUTOMOTIVE

Gasto	Costo
Ensamble COLCIRCUITOS	\$437.000
Componentes esquema, más gastos de envíos.	\$200.000
Total	\$637.000

4.1 MODOS DE FUNCIONAMIENTO DE LA TARJETA ARGUS AUTOMOTIVE

El funcionamiento del dispositivo depende de dos switch o jumpers que están implementados en el esquema general del proyecto, con los nombres JUMPER1 (JP1) y JUMPER2 (JP2), mostrados en la Figura 43. El modo con el que trabaja la tarjeta depende de que jumper este cerrado.

Figura 43. Ubicación de los jumpers en el dispositivo



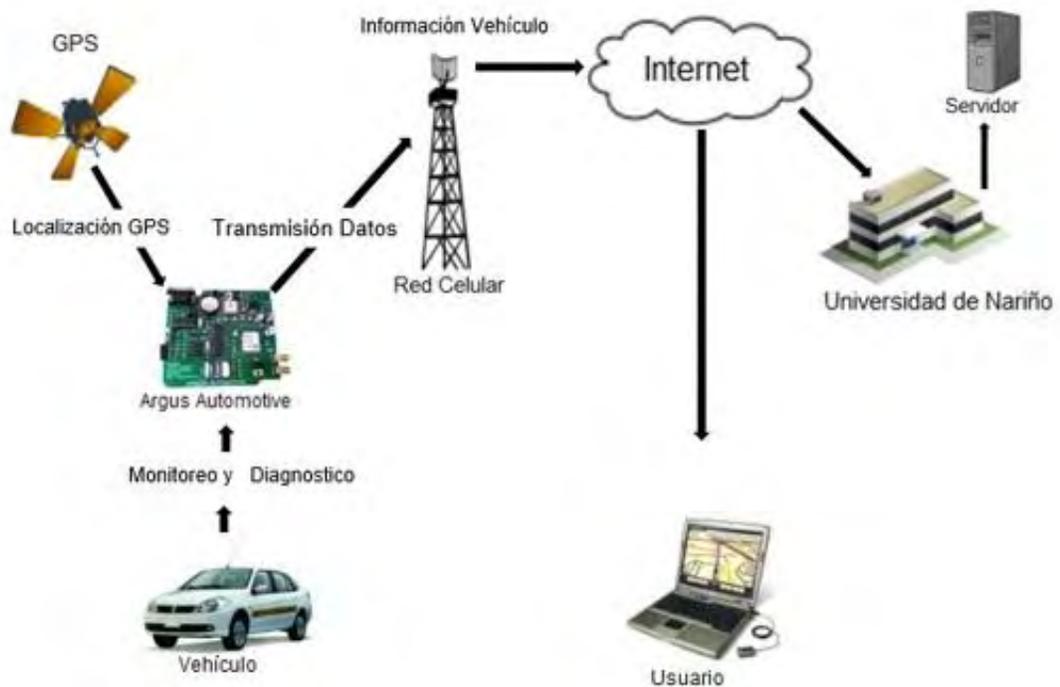
- Modo Programación: este modo permite la programación de ARGUS AUTOMOTIVE con el programa del proyecto, que permite monitorear y diagnosticar cualquier vehículo a larga distancia. Para esto ningún jumper debe estar cerrado.
- Modo Prueba: este modo es usado para probar el funcionamiento de la tarjeta conectándola a un computador y al mismo tiempo al vehiculo, en este se puede apreciar la posición del GPS, además de datos adquiridos del vehículo directamente en el puerto serial de ARDUINO, sin necesidad de abrir la WEB. Para este modo únicamente el jumper JP1 debe estar cerrado.
- Modo Funcionamiento: es el modo con el que el dispositivo trabaja ya en el vehículo, una vez ha sido programado y probado. Para este modo JP1 y JP2 deben estar cerrados.

5. FIRMWARE ARGUS AUTOMOTIVE

El firmware de la tarjeta LM8095 como también se la ha bautizado tiene diferentes etapas en su programación interna, ya que el dispositivo realizado es capaz de adquirir y transmitir datos generados por el automóvil hacia cualquier servidor para monitorear y diagnosticar posibles fallas que se puedan presentar en el vehículo.

Esto se logra con base en la red electrónica CAN Bus de los automóviles modernos de donde se obtienen todos los datos que se desean para cumplir los objetivos de este proyecto. Además, el dispositivo también procesa los datos adquiridos desde el sistema de posicionamiento global GPS, y una vez el microcontrolador de ARGUS AUTOMOTIVE obtiene estas tramas, las envía a un servidor de la Universidad de Nariño por medio de internet y gracias al subsistema de comunicación GSM/GPRS. La Figura 44 muestra el modelo del sistema del funcionamiento de la tarjeta ARGUS AUTOMOTIVE.

Figura 44. Modelo del sistema del funcionamiento de ARGUS AUTOMOTIVE



Como se aprecia en el modelo del funcionamiento de ARGUS AUTOMOTIVE, este interactúa con diferentes sistemas que presentan características propias en especial los automóviles, ya que existen variedad de modelos y marcas, y a pesar de que todos los automóviles que se rigen con el protocolo OBD-II y con CAN Bus son estándares, puede variar la velocidad con la que el bus trabaja en los vehículos, la gran mayoría de carros trabaja con una velocidad bus de 500 Kbps, por lo que el dispositivo está programado por defecto con esta velocidad, pero también hay automóviles que trabajan con velocidades de 10 Kbps, 125Kbps, 250Kbps y 1Mbps.

Otro aspecto importante es la transmisión de datos, la cual se realiza con un tiempo de muestreo de 5 segundos, tiempo que si se desea puede cambiarse en el firmware del dispositivo.

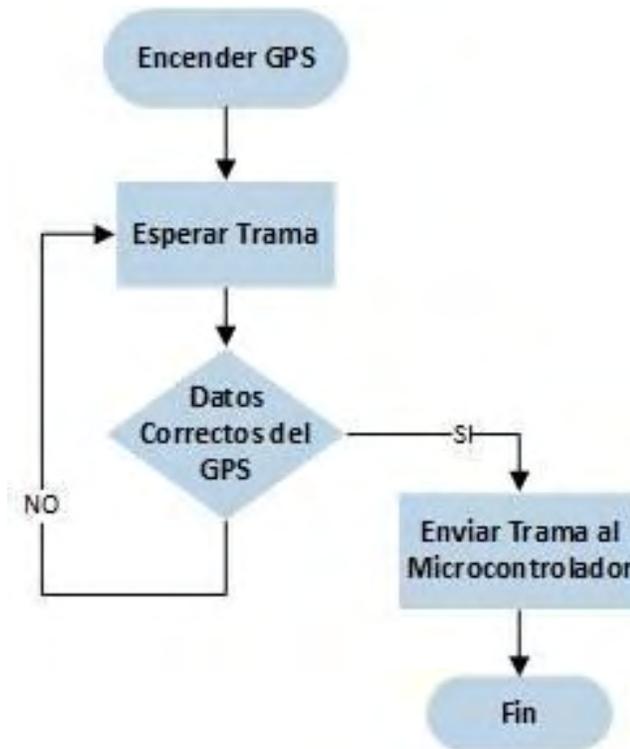
Por otra parte, ARGUS AUTOMOTIVE no cuenta con un dispositivo de almacenamiento interno, los datos que se envían al servidor y se guardan en un archivo de texto o base de datos, son el único registro de actividad que se almacena, el tamaño del archivo de texto depende del servidor, el utilizado actualmente brinda una capacidad aproximada a 4 GB, la cual es muy grande considerando que el dispositivo en aproximadamente 4 meses de uso a almacenado apenas 160 KB.

5.1 FIRMWARE PARA OBTENER POSICIÓN GPS

Como se mostró en el marco teórico, el módulo GPS Quectel L80 brinda muchas tramas con diferente información. Para este proyecto Se utiliza dos de estas. La trama GPGGA y GPRMC, que brindan datos como longitud, latitud, velocidad, hora, y fecha, información suficiente para establecer la posición del automóvil en el que se encuentra el dispositivo.

En el diagrama de flujo de la Figura 45, se muestra la descripción del proceso para que el GPS otorgue el posicionamiento.

Figura 45. Diagrama de flujo del firmware para el posicionamiento del GPS



Para la implementación de esta rutina se utilizaron bibliotecas disponibles en la plataforma ARDUINO, lo cual se realizó en lenguaje C/C++.

5.2 FIRMWARE PARA MONITOREO Y DIAGNOSTICO

El firmware para CAN Bus se divide en dos segmentos: el primero que se refiere a la parte de monitoreo, y el segundo a la de diagnóstico.

5.2.1 Firmware Monitoreo de Variables

Para la adquisición de datos a monitorear en el automóvil primero se debe hacer una configuración en la tarjeta definiendo los baudios de velocidad a la cual se va a comunicar con el auto. Posterior a esto, se define una función en donde se hace una solicitud del estado de las variables a monitorear. En la solicitud se envía una trama de datos en la que se especifica el modo de operación de la tarjeta ARGUS AUTOMOTIVE. El estándar OBDII define ocho modos de operación, los que

permiten monitoreo, diagnóstico, solicitud de información del automóvil, borrar datos de fallas, entre otros. Cabe destacar que para hacer el monitoreo de las variables, el modo de operación es el Modo 02 que muestra información de un evento almacenado. Además de esto, la trama está compuesta por el parámetro de identificación ID de cada variable a monitorear, por lo que para saber el estado de las variables se debe enviar una trama específica para cada una. La Tabla 14 detalla los ID de los parámetros monitoreados en el presente proyecto.

Tabla 14. Variables a monitorear del automóvil con su respectivo ID

Variable a Monitorear	ID
Velocidad	0D
Temperatura del líquido refrigerante	05
Revoluciones por minuto	0C
Posición del acelerador	11
Porcentaje de carga del motor	04

Fuente: Wikipedia

URL: https://es.wikipedia.org/wiki/OBD-II_PID

Enseguida a esto se debe esperar el mensaje de respuesta de la red CAN Bus, que consiste en una trama de datos con los valores respectivos de cada variable, que posteriormente es analizada y procesada. La trama se compone de 4 bytes denominados A, B, C y D, en cada byte se enumeran los bit de 0 a 7, siendo el 7 el bit más significativo y el 0 es el bit menos significativo. En la Figura 46 se ve como está compuesta la trama.

Figura 46. Formato de la trama respondida por la red CAN BUS

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

Cada variable monitoreada, tiene una fórmula específica para su cálculo, y una unidades ya establecidas. La Tabla 15 muestra las variables del vehículo, las unidades en las que se mide cada variable y las fórmulas para sus cálculos exactos, en las cuales se debe reemplazar los valores de los bytes A, B, C y D que se obtienen de la trama de respuesta generada por el automóvil.

Tabla 15. Parámetros de las variables de monitoreo

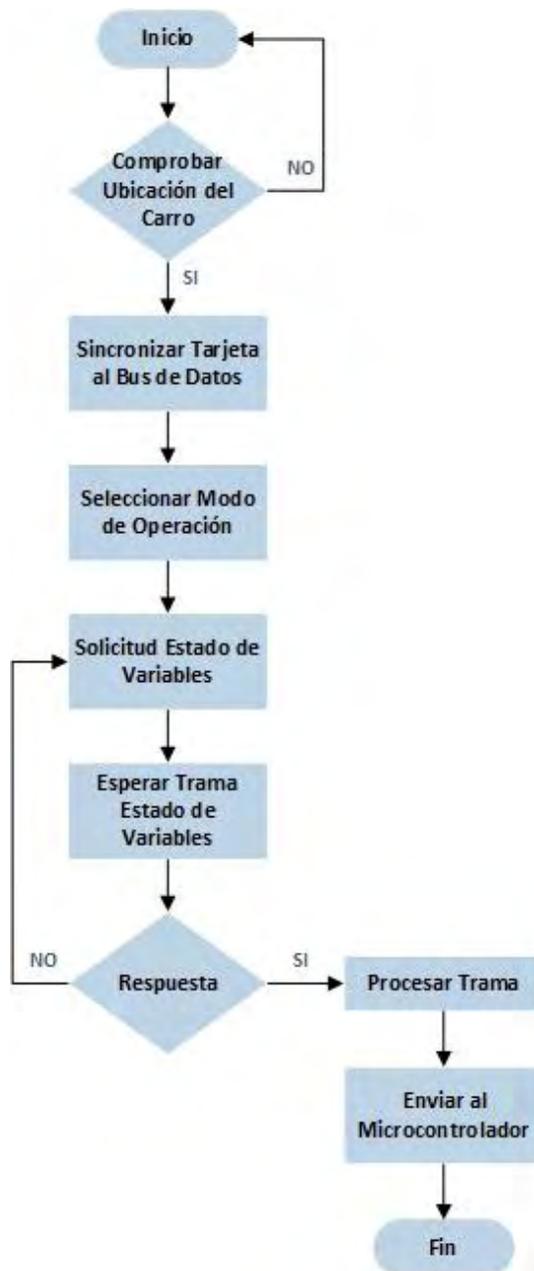
ID	Variable	Bytes de respuesta	Unidades	Formula
0D	Velocidad	A	Km/h	A
05	Temperatura del líquido refrigerante	A	°C	A-40
0C	Revoluciones por minuto	A,B	rpm	$\frac{256A + B}{4}$
11	Posición del acelerador	A	%	$\frac{100}{250} * A$
04	Porcentaje carga motor	A	%	$\frac{100}{250} * A$

Fuente: Wikipedia

URL: https://es.wikipedia.org/wiki/OBD-II_PID

La Figura 47 muestra el diagrama del flujo del proceso anteriormente nombrado para el monitoreo del automóvil.

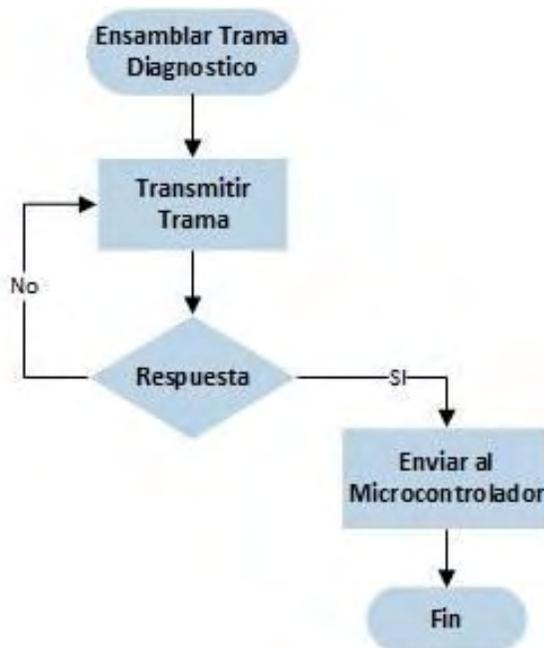
Figura 47. Diagrama de flujo para el monitoreo del automóvil



5.2.2 Firmware de Diagnóstico

Para hacer el diagnóstico de las fallas presentes en el automóvil se debe hacer una solicitud por medio de una cadena de caracteres, en donde se especifica el modo de operación que para este caso es Modo 3. Posterior a esto, se espera la respuesta del auto y este resultado se ensambla en la trama general. El diagrama de flujo de la Figura 48 muestra el procedimiento.

Figura 48. Diagrama de flujo para el diagnóstico del automóvil

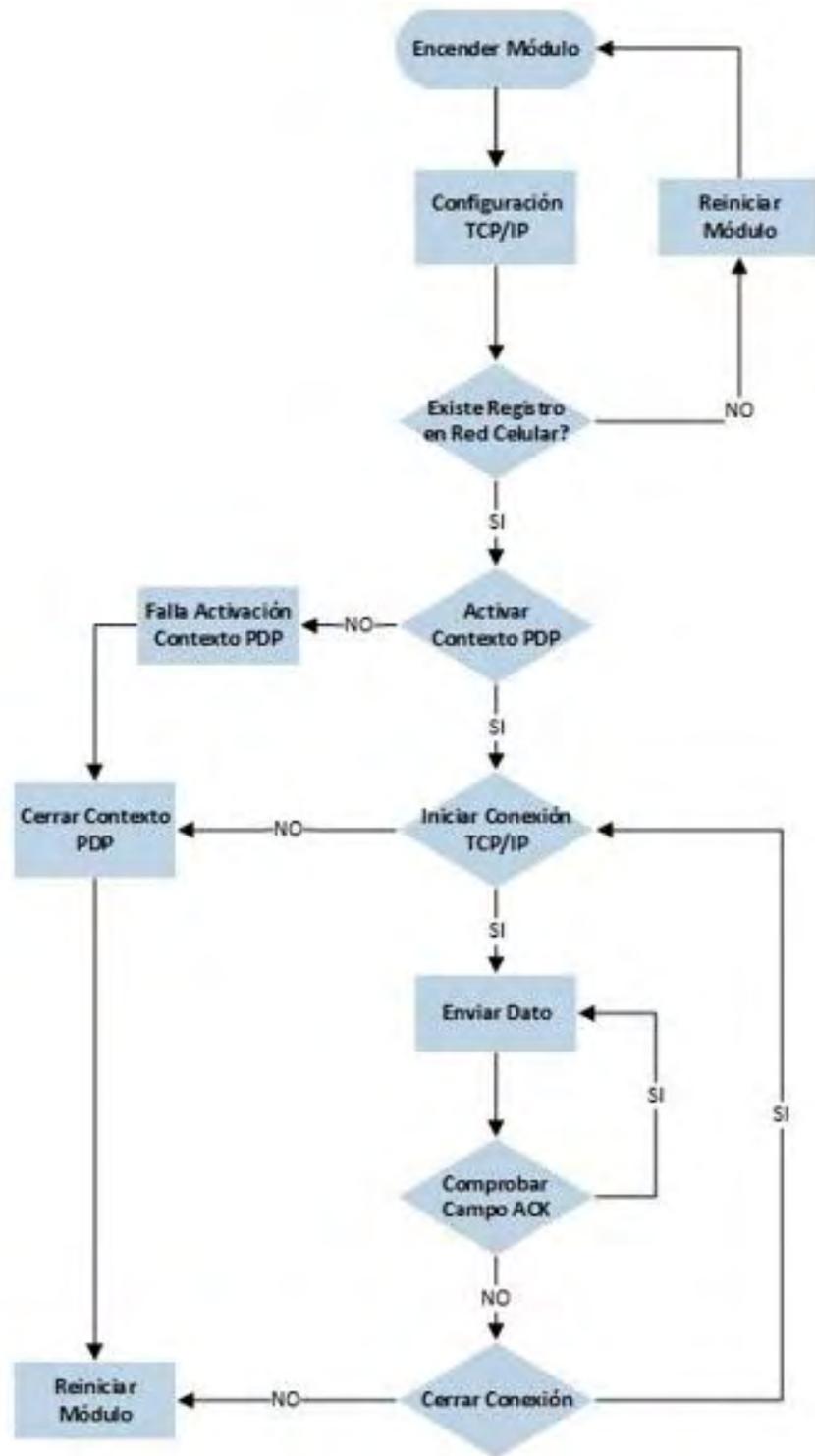


5.3 FIRMWARE PARA EL ENVIO DE DATOS GPRS

Una vez se tenga la posición exacta y los datos a monitorear del vehículo, la tarjeta LM8095 está preparada para la transmisión de la información hasta el servidor. Para el funcionamiento del dispositivo Quectel M95 se definen los pines de encendido y estado, junto con los pines de transmisión y recepción de datos (TX y RX) que se conectan con el microcontrolador ATmega328P-PU. Posteriormente se realizan especificaciones para el encendido del dispositivo, además de las funciones necesarias para su correcto funcionamiento, como registrar el módulo en la red, definir protocolo de datos, indicar el punto de acceso APN, activar contexto GPRS, entre otras, las cuales se basan en comando AT. La

Figura 49 muestra el diagrama de flujo que permite que el módulo Quectel M95 envíe datos al servidor.

Figura 49. Diagrama de flujo para la transmisión de datos



Después de prender el dispositivo por medio de pines digitales del microcontrolador, todo el firmware del módulo GSM/GPRS, se fundamenta en comandos AT. Algunos de las instrucciones más importantes y usadas en el presente proyecto, en orden son las siguientes:

- AT+CREG?: Averigua si el módulo ya se registró en la red celular.
- AT+QIFGCNT=0: Define contexto PDP o protocolo de datos por paquetes.
- AT+QICSGP=1: Indica el nombre de punto de acceso APN (Access Point Name) en este caso internet.movistar.com.co, para que el dispositivo pueda acceder a una red de datos basada en GPRS.
- AT+QIREGAPP: Inicia conexión con el protocolo de control de transmisión TCPIP (Transmission Control Protocol) y establece el APN.
- AT+QIACT: Activa el contexto GPRS.
- AT+QPING: Testea un servidor remoto, en este caso el servidor de ingeniería de la Universidad de Nariño.
- AT+QIOPEN: Inicia conexión con el servidor y con el protocolo TCP, otorgándole al dispositivo dirección IP.
- AT+QISEND: Comando que envía las tramas de datos a través de la conexión TCP.
- AT+QICLOSE: Cerrar la conexión TCP, terminando con la transmisión de datos.

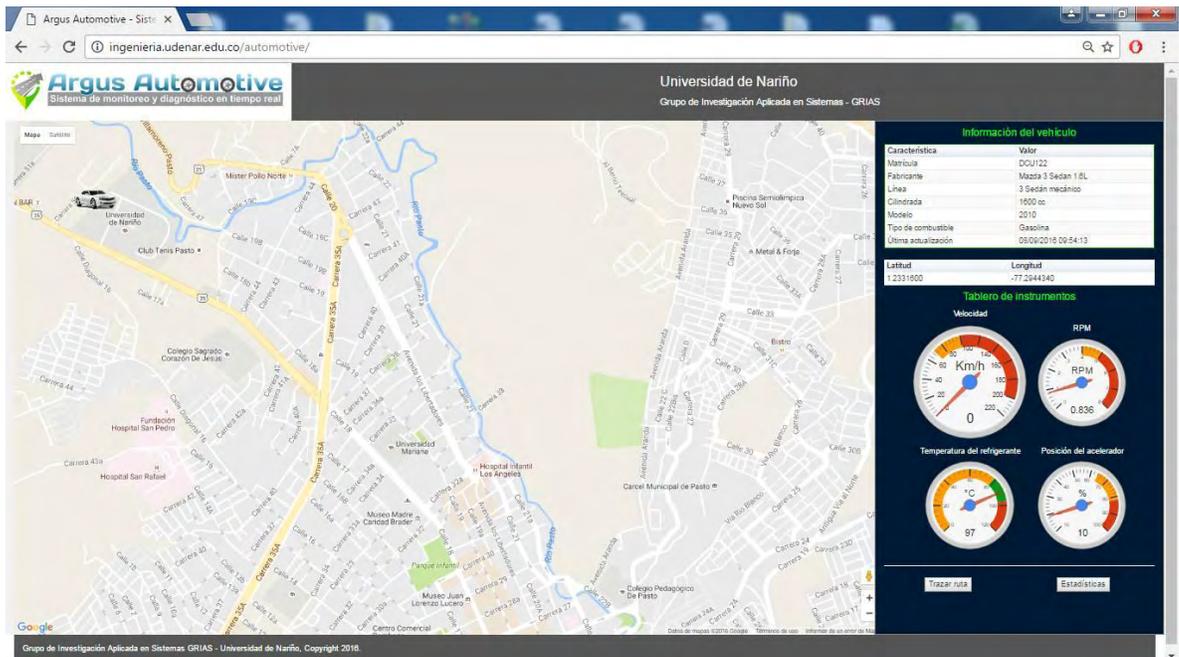
Con el anterior proceso se realiza el envío de datos obtenidos con la tarjeta ARGUS AUTOMOTIVE del automóvil, como es la ubicación exacta del vehículo, las variables de monitoreo, los datos son recibidos por el servidor y se almacenan en un archivo que posteriormente es procesado por una página web. Una de las características adicionales que tiene el dispositivo, es la interacción con el celular del usuario. Esto se hace a través del envío de un mensaje de texto, en el cual se informa cuando la tarjeta ha perdido conexión con el servidor. Una vez se restablezca esta conexión, ARGUS AUTOMOTIVE también lo informa por medio de otro mensaje de texto. Todo este proceso fue realizado por el Mg Juan Carlos Castillo Eraso como parte del desarrollo del proyecto de investigación.

6. RESULTADOS

A continuación se describe las pruebas que se realizaron con el dispositivo ARGUS AUTOMOTIVE LM8095, con el fin de analizar si el dispositivo cumple con los requerimientos que se habían establecido previamente. Estas pruebas se realizaron en un automóvil de la marca Mazda modelo Mazda 3 Sedan.

Los resultados obtenidos tanto de la posición del automóvil como los datos de monitoreo se pueden observar a través de la página web de URL <http://ingenieria.udenar.edu.co/automotive/> creada por el Mg Juan Carlos Castillo Eraso, la cual se muestra en la Figura 50.

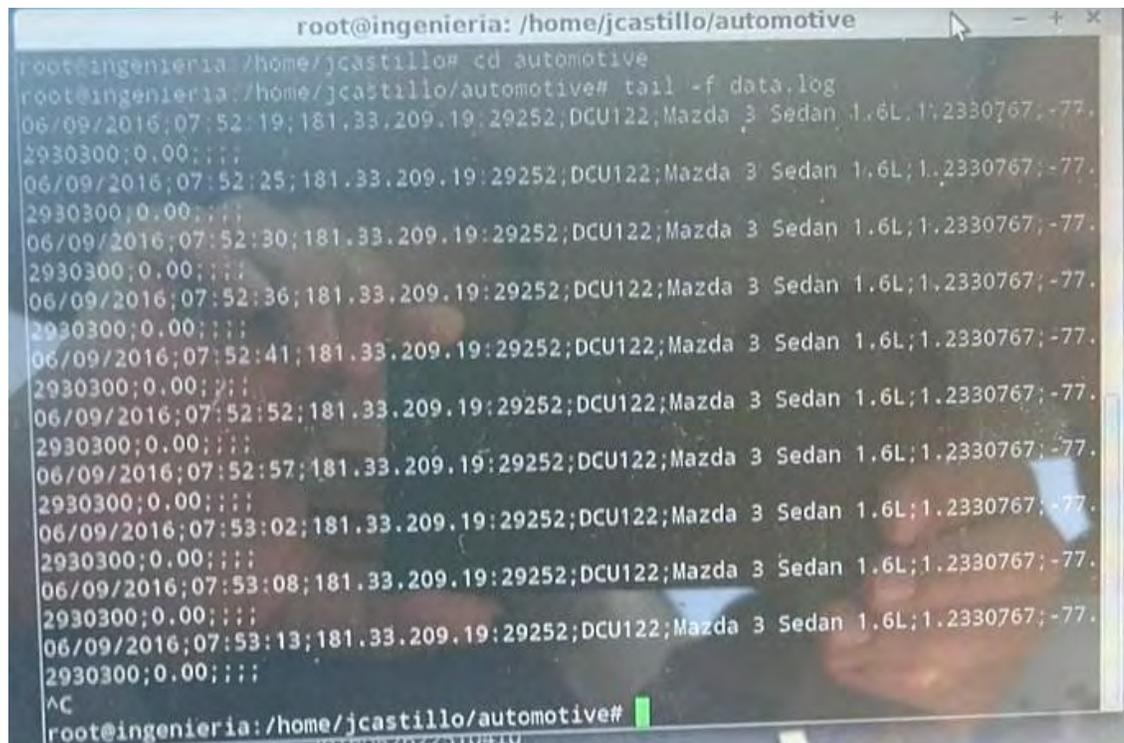
Figura 50. Página web del dispositivo



6.1 PRUEBA DE COMUNICACIÓN CON EL SERVIDOR

En esta prueba se muestra los datos que recibe el servidor, como se puede observar en la Figura 51. La trama de datos está compuesta por la fecha de cuando el dispositivo se conectó al carro, los datos recibidos del monitoreo y diagnóstico, las coordenadas de ubicación del mismo, y por último el modelo del automóvil y su respectiva placa.

Figura 51. Recepción de los datos del automóvil en el servidor

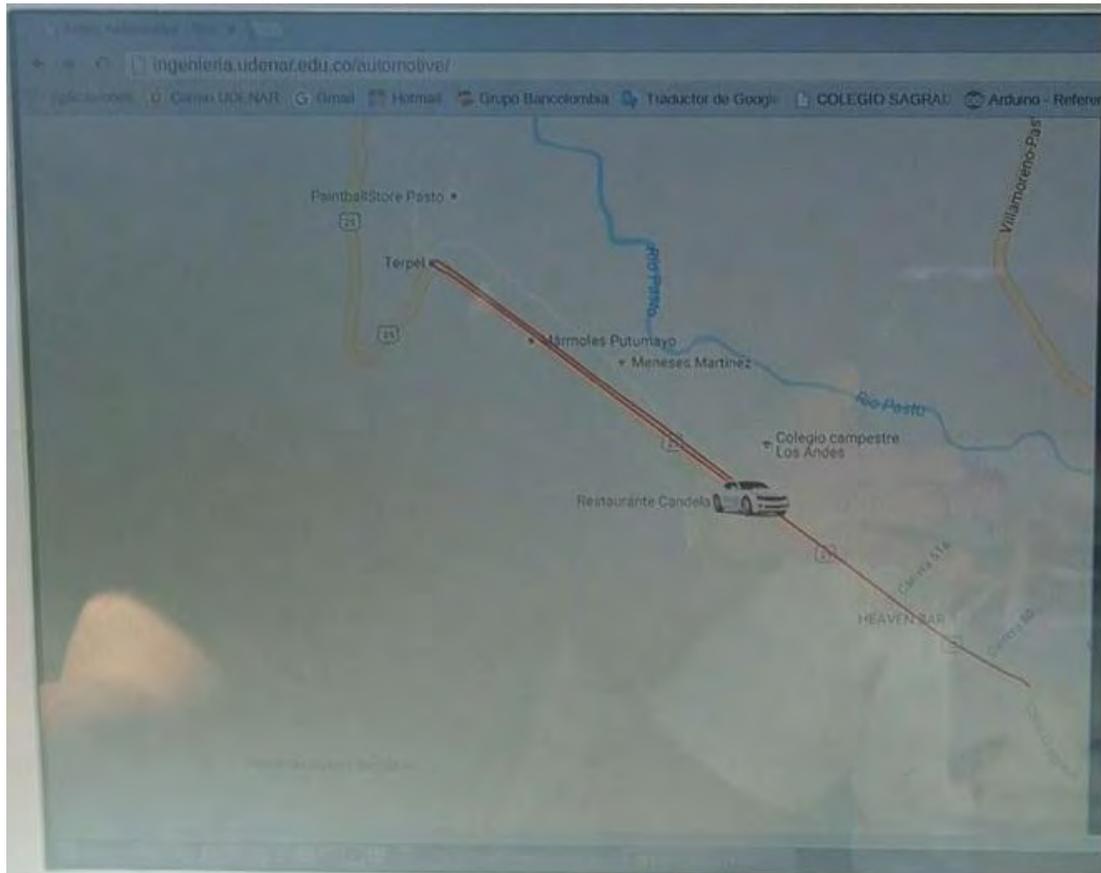
A terminal window titled 'root@ingenieria: /home/jcastillo/automotive' showing the execution of 'tail -f data.log'. The output displays a series of log entries, each containing a timestamp, IP address, port, location coordinates, car model, and license plate. The entries are: '06/09/2016;07:52:19;181.33.209.19:29252;DCU122;Mazda 3 Sedan 1.6L;1.2330767;-77.2930300;0.00;;;', '06/09/2016;07:52:25;181.33.209.19:29252;DCU122;Mazda 3 Sedan 1.6L;1.2330767;-77.2930300;0.00;;;', '06/09/2016;07:52:30;181.33.209.19:29252;DCU122;Mazda 3 Sedan 1.6L;1.2330767;-77.2930300;0.00;;;', '06/09/2016;07:52:36;181.33.209.19:29252;DCU122;Mazda 3 Sedan 1.6L;1.2330767;-77.2930300;0.00;;;', '06/09/2016;07:52:41;181.33.209.19:29252;DCU122;Mazda 3 Sedan 1.6L;1.2330767;-77.2930300;0.00;;;', '06/09/2016;07:52:52;181.33.209.19:29252;DCU122;Mazda 3 Sedan 1.6L;1.2330767;-77.2930300;0.00;;;', '06/09/2016;07:52:57;181.33.209.19:29252;DCU122;Mazda 3 Sedan 1.6L;1.2330767;-77.2930300;0.00;;;', '06/09/2016;07:53:02;181.33.209.19:29252;DCU122;Mazda 3 Sedan 1.6L;1.2330767;-77.2930300;0.00;;;', '06/09/2016;07:53:08;181.33.209.19:29252;DCU122;Mazda 3 Sedan 1.6L;1.2330767;-77.2930300;0.00;;;', and '06/09/2016;07:53:13;181.33.209.19:29252;DCU122;Mazda 3 Sedan 1.6L;1.2330767;-77.2930300;0.00;;;'. The terminal ends with '^C' and a prompt 'root@ingenieria: /home/jcastillo/automotive#'.

```
root@ingenieria: /home/jcastillo/automotive
root@ingenieria /home/jcastillo# cd automotive
root@ingenieria: /home/jcastillo/automotive# tail -f data.log
06/09/2016;07:52:19;181.33.209.19:29252;DCU122;Mazda 3 Sedan 1.6L;1.2330767;-77.
2930300;0.00;;;
06/09/2016;07:52:25;181.33.209.19:29252;DCU122;Mazda 3 Sedan 1.6L;1.2330767;-77.
2930300;0.00;;;
06/09/2016;07:52:30;181.33.209.19:29252;DCU122;Mazda 3 Sedan 1.6L;1.2330767;-77.
2930300;0.00;;;
06/09/2016;07:52:36;181.33.209.19:29252;DCU122;Mazda 3 Sedan 1.6L;1.2330767;-77.
2930300;0.00;;;
06/09/2016;07:52:41;181.33.209.19:29252;DCU122;Mazda 3 Sedan 1.6L;1.2330767;-77.
2930300;0.00;;;
06/09/2016;07:52:52;181.33.209.19:29252;DCU122;Mazda 3 Sedan 1.6L;1.2330767;-77.
2930300;0.00;;;
06/09/2016;07:52:57;181.33.209.19:29252;DCU122;Mazda 3 Sedan 1.6L;1.2330767;-77.
2930300;0.00;;;
06/09/2016;07:53:02;181.33.209.19:29252;DCU122;Mazda 3 Sedan 1.6L;1.2330767;-77.
2930300;0.00;;;
06/09/2016;07:53:08;181.33.209.19:29252;DCU122;Mazda 3 Sedan 1.6L;1.2330767;-77.
2930300;0.00;;;
06/09/2016;07:53:13;181.33.209.19:29252;DCU122;Mazda 3 Sedan 1.6L;1.2330767;-77.
2930300;0.00;;;
^C
root@ingenieria: /home/jcastillo/automotive#
```

6.2 PRUEBA DE UBICACIÓN DEL AUTOMÓVIL

En esta prueba se puede observar que el sistema de ubicación que se le incorporó al dispositivo funciona correctamente y que tiene buena precisión, dando la ubicación exacta por donde se está desplazando el auto. Un ejemplo se ilustra en la Figura 52.

Figura 52. Ubicación del automóvil en el mapa



6.3 PRUEBA DE DATOS ADQUIRIDOS DEL AUTOMÓVIL

En la Figura 53 se puede observar un tablero de instrumentos donde se indica el estado de las variables de monitoreo. Cabe destacar que cuando se realizó esta prueba los valores se los compararon con los presentes en el cuadro de instrumentos del automóvil y se pudo observar que eran completamente iguales.

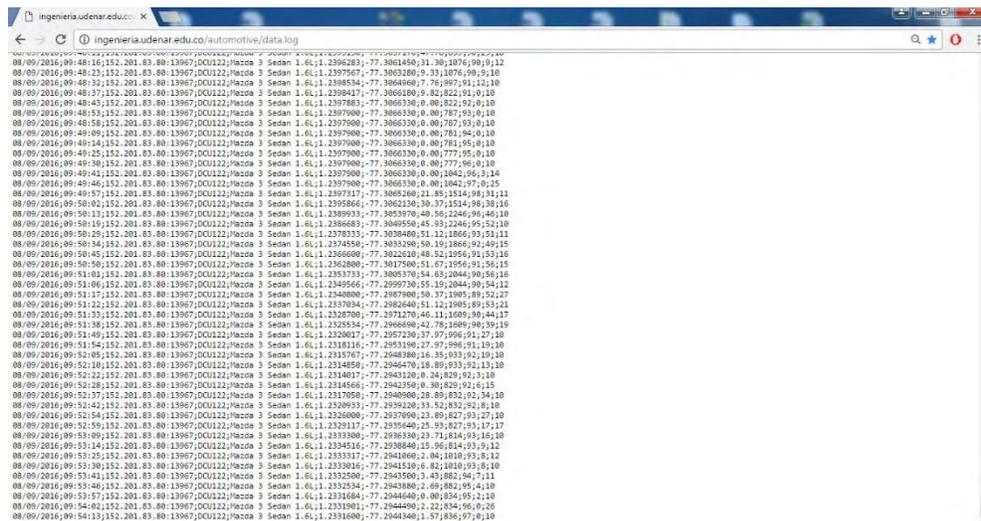
Figura 53. Tablero de instrumentos en página web



6.4 BASE DE DATOS ADQUIRIDOS POR ARGUS AUTOMOTIVE

Todos los datos adquiridos por el dispositivo se almacenan en un archivo de texto, como se observa en la Figura 54. Esta información se la puede descargar a través del URL: <http://ingenieria.udenar.edu.co/automotive/data.log>.

Figura 54. Base de datos



La Figura 55 detalla la estructura de cada una de las tramas que se mandan a la base de datos.

Figura 55. Estructura de una trama en la base de datos

08/09/2016;09:48:23;152.201.83.80:13967;DCU122;Mazda 3 Sedan 1.6L;1.2397567;-77.3063280;9.33;1076;90;9;10;14;0

- Año y hora en que la trama fue recibida por el servidor.
- IP y puerto asignado por el servidor al dispositivo.
- Placa, modelo y cilindraje del automóvil monitoreado (estos datos son ingresados manualmente en el software del dispositivo).
- Latitud y longitud de la ubicación del módulo GPS.
- Velocidad que otorga la trama del GPS.
- Variables de monitoreo del automóvil en orden (rpm, temperatura del líquido refrigerante, velocidad, posición del acelerador, carga del motor).
- Diagnóstico del automóvil.

CONCLUSIONES

La motivación para el desarrollo de este proyecto surge a partir de la falta de información en tiempo real del vehículo cuando este tiene un trayecto cotidiano, además de la necesidad de un sistema para la ubicación del mismo en caso de robo.

Aunque en el mercado existen diferentes tipos de instrumentos para realizar un diagnóstico automotriz, estos carecen de algunas de las características con las que cuenta ARGUS AUTOMOTIVE, ya que con esta tarjeta además del diagnóstico se puede obtener un monitoreo del estado de algunas variables, la ubicación del automóvil. Una de las características más relevantes con las que cuenta esta tarjeta es que toda la información recopilada es enviada a través de la red celular a un servidor. Lo que es importante para la vigilancia a larga distancia de cualquier automóvil que se desee controlar, siendo útil para evitar posibles siniestros por fallas en el auto.

Con la implementación de la tarjeta ARGUS AUTOMOTIVE el usuario economizaría gastos a la hora de saber sobre el estado de su vehículo, ya que en los centros automotrices esta labor es muy costosa.

El módulo GPS QUECTEL L80 es preciso marcando la posición del vehículo con un radio de error de máximo 5 metros, pero puede llegar a tardar aproximadamente 5 minutos en ubicarse en ambientes que se encuentran demasiado cerrados.

El módulo GSM/GPRS QUECTEL M95, tiene un excelente funcionamiento, el problema es la etapa de acople que se debe diseñar para que pueda trabajar con dispositivos de diferente voltaje de alimentación, seguramente otro módulo GPRS no presente este inconveniente.

Lastimosamente, a pesar de que ARGUS AUTOMOTIVE es capaz de otorgar al servidor la gran mayoría de códigos de diagnóstico, en muchos casos es difícil saber sus significados o las fallas a las que hacen referencia, ya que algunas empresas utilizan códigos privados, que no son de acceso al público. Pero también existen unos códigos genéricos los cuales la tarjeta los puede diagnosticar e informar al usuario sobre la falla que se está presentando en su vehículo.

RECOMENDACIONES

El dispositivo ARGUS AUTOMOTIVE LM8095 está diseñado en doble capa, y a pesar de que su tamaño es pequeño, con una mayor inversión económica se podría realizar un prototipo con un mayor número de capas, lo que reduciría sus dimensiones notablemente.

A pesar de que el microcontrolador utilizado ATmega328P-PU fue suficiente para el desarrollo de este proyecto, sería interesante reemplazarlo por un microcontrolador de mejores características y mayor capacidad de memoria, para que así el dispositivo pueda aumentar variables de monitoreo y diagnóstico.

El Quectel M95 módulo GSM/GPRS, permite muchas más opciones de interfaz que en este proyecto no fueron utilizadas, pero que sería una propuesta atractiva si se quisiera mejorar esta versión de la tarjeta, como aumentar micrófono y audífonos, para mantener un contacto auditivo con el automóvil desde cualquier computador.

A pesar de que en los objetivos de este proyecto no se planteó el diseño de una página WEB para mirar los resultados, este proceso podría ser mejorado para monitorear y diagnosticar más variables.

Antes de conectar la tarjeta LM8095 a cualquier vehículo, es recomendable investigar sobre la velocidad a la cual trabaja el bus de datos del automóvil, para sincronizar la tarjeta a la misma velocidad y tener un óptimo funcionamiento.

La validación del sistema del dispositivo podría ser mejorada, ya que la única forma que se tiene actualmente para saber si los datos obtenidos con la tarjeta son verídicos, es la comparación visual de estos con el cuadro de instrumentos presente en el vehículo.

El almacenamiento de los datos obtenidos del vehículo se realiza en un archivo de texto, sin embargo el dispositivo no cuenta con almacenamiento interno, por lo que se podría mejorar el diseño implementando un sistema de almacenamiento de información a través de una Micro SD.

REFERENCIAS

- [1] L. Letham, «GPS FÁCIL Uso del Sistema de Posicionamiento Global,» Barcelona, Paidotribo, 2001, p. 5.
- [2] Ó. Torrente Artero, «ARDUINO Curso Práctico de Formación,» Madrid, RC libros, 2013, pp. 62,64.
- [3] I. Orte Ruiz, «LEDSRIPT,» de *Proyecto Final de Carrera*, Universidad Politécnica de Valencia, Valencia, 2013, p. 13.
- [4] J. I. Espinoza Armijo, «Dimensionamiento de la Capacidad y Caracterización de la Calidad de una Red GPRS,» de *Proyecto Final de Carrera*, Universidad de Chile, Santiago de Chile, 2007, pp. 7-9.
- [5] A. R. Reinoso García y C. A. Tocaín Villarruel, «Diseño de un Prototipo para Controlar un Semáforo Inteligente Usando Tecnologías GSM/GPRS y Wireless CPU Sobre una Plataforma OPEN-SOFT(Linux),» de *Proyecto Final de Carrera*, Escuela Politécnica Nacional, Quito, 2009, pp. 38-49.
- [6] R. Sánchez Vítors, «Técnica Industrial Especial Electricidad y Electronica, Aplicaciones de los Sistemas Embebidos,» Octubre 2004. [En línea]. Available: <http://www.tecnicaindustrial.es/tiadmin/numeros/15/07/a07.pdf>. [Último acceso: Marzo 2016].
- [7] E. Chillaron i Farré, «Introducción al Sistema GPRS y a su gestión de Recursos Radio,» *Buran*, nº 19, p. 38, 2003.
- [8] I. Sedano, «Tutorial de GPRS,» *eRemoteControl - Sistemas de telecontrol*, nº 1, p. 4, 2013.
- [9] F. D. J. Castillo Ramírez, G. Martínez Estrella y E. Mejía Ramírez, «Diseño de un Sistema Detector de Intrusos con Alertamiento Vía Celular, para Casa Habitación en el Estado de México,» de *Proyecto Final de Carrera*, Instituto Politécnico Nacional, México, 2010, p. 43.
- [10] Quectel Wireless Module Expert, «Quectel M95 AT Commands Manuel,» 16 Julio 2013. [En línea]. Available: http://www.quectel.com/UploadImage/Downlad/M95_AT_Commands_Manual_V1.2.pdf.

[Último acceso: Marzo 2016].

- [11] J. D. Pañi Arévalo y P. G. Pinos Guartamber , «Compendio del Sistema de Transmision de Datos CAN-BUS,» de *Proyecto Final de Carrera*, Universidad Poitécnica Salesiana, Cuenca, 2007, pp. 2-5.
- [12] T. A. Tenezaca Arpi, «Estudio para Simular una Red CAN con Aplicación en comunicación de dispositivos electrónicos en el automóvil,» de *Proyecto Final de Carrera*, Universidad del Azuay, Cuenca, 2013, pp. 8-27.
- [13] J. Calva Cuenca, «Diseño e Implementación de Protocolo CAN para el control de un Módulo de Red de Sensores,» de *Proyecto Final de Carrera*, Escuela Politécnica Nacional, Quito, 2010, pp. 16-22.
- [14] M. L. Defaz Andrango, «Estudio del Protocolo CAN (Controller Area Network) y su Aplicación en Redes de Control,» de *Proyecto Final de Carrera*, Escuela Politécnica Nacional, Quito, 2007, p. 34.
- [15] H. . N. Alepus Soriano , H. Giménez Espinoza De Los Montero, J. M. Jordá Fernández, F. Ripoll Gisbert, P. M. Saval Sendra , J. Serrano Reche y J. V. Talens Noguera , «BUS CAN,» [En línea]. Available: <http://server-die.alc.upv.es/asignaturas/PAEEES/2005-06/A03-A04%20-%20Bus%20CAN.pdf>. [Último acceso: Marzo 2016].
- [16] J. A. López Fresno, «Nodo de Comunicación Basado en el Bus CAN,» de *Proyecto Final de Carrera*, Universitat Rovira I Virgili, 2004, p. 20.
- [17] I. D. Villamar Aquirre, «Estudia y Análisis de los Sistemas de Diagnóstico en los Automóviles Modernos, Sistemas OBDII,» de *Proyecto Final de Carrera*, Universidad del Azuay , Cuenca, Ecuador, 2008, p. 8.
- [18] F. E. Guerrero Peralvo, «Diseño y Construcción del Banco de Pruebas para Comprobación y Verificación de Computadores Automotrices con Inyección Electronica a Gasolina,» de *Proyecto Final de Carrera*, Escuela Politécnica del ejército, Latacunga, 2006.
- [19] I. Quintanilla García, «NMEA National Marine Electronic Asociation [Archivo de video],» 28 Diciembre 2009. [En línea]. Available: <https://riunet.upv.es/handle/10251/6806>. [Último acceso: Marzo 2016].
- [20] Quectel Wireless Module Expert, «Quectel L80 Protocol Specification,» Junio 2013. [En línea]. Available: <http://www.quectel.com/es/product/prodetail.aspx?id=62>. [Último acceso: Marzo 2016].
- [21] J. J. Zambrano Termal, «Prototipo de Comunicación Entre Taxis-Policía para la Seguridad Ciudadana,» de *Proyecto Final de Carrera* , Universidad San Buenaventura, Santiago de Cali,

2014, p. 34.

- [22] Argos, «Nomadas Electronicos,» 25 Febrero 2015. [En línea]. Available: <https://nomadaselectronicos.wordpress.com/2015/02/25/traductor-de-niveles-logicos-de-voltaje-level-translator/>. [Último acceso: Marzo 2016].
- [23] F. J. Alvear Granja y J. F. Hernando, «Desarrollo de una Interfaz USB-CAN,» de *Proyecto Final de Carrera*, Universidad de Valladolid, Valladolid, 2012, p. 4.
- [24] P. R. Mejía Morales , J. L. Poma Montaña y J. L. Ramón Pineda, «Diseño y Construcción una Interfaz Didactica de Redes y Multiplexado CAN para Aplicaciones en el Automóvil,» de *Proyecto Final de Carrera*, Universidad Politécnica Salesiana , Cuenca, 2013, p. 79.
- [25] Y. L. Leon Vanegas y M. A. Sanches Cardenas, «Años potencialmente perdidos por accidentes de tránsito, Colombia 2010,» *CES Salud Pública*, vol. II, nº 2, pp. 159-168, 2011.
- [26] N. Semiconductor, «Electronic Components Datasheet Search,» Junio 2004. [En línea]. Available: <http://pdf1.alldatasheet.com/datasheet-pdf/view/8606/NSC/LM1117T-3.3.html>. [Último acceso: Marzo 2016].
- [27] Quectel Wireless Module Expert, «Quectel L80,» 2015. [En línea]. Available: http://www.quectel.com/UploadFile/Product/Quectel_L80_GPS_Specification_V1.2.pdf. [Último acceso: Marzo 2016].
- [28] Quectel Wireless Module Expert, «Quectel L80 Hardware Design,» Agosto 2013. [En línea]. Available: http://www.quectel.com/uploadimage/download/l80_hardware_design_v1.1.pdf. [Último acceso: Marzo 2016].
- [29] I. Alab, L. Pereira, P. Braga, R. Azevedo, A. Sena, R. Hidaka, L. Luz, M. Moscoso, I. Teixeira, S. Bermeguy, T. Teixeira y L. Vieira, «Aplicativo Controlador de Portões,» IESAM, Belém, 2013.
- [30] Atmel, «ATmega328,» Noviembre 2015. [En línea]. Available: <http://www.atmel.com/devices/atmega328.aspx>. [Último acceso: Marzo 2016].
- [31] Ó. Torrente Artero, «ARDUINO Curso Práctico de Formación,» Madrid, RC libros, 2013, p. 64.
- [32] Quectel Wireless Module Expert, «Quectel M95,» 2014. [En línea]. Available: http://www.quectel.com/UploadFile/Product/Quectel_M95_GSM_Specification_V3.0.pdf. [Último acceso: Marzo 2016].
- [33] Quectel Wireless Module Expert, «Quectel M95 Hardware Design,» Septiembre 2013. [En línea]. Available: http://www.quectel.com/UploadImage/Downlad/M95_Hardware_Design_V1.3.pdf. [Último acceso: Marzo 2016].

- [34] ACME Systems, [En línea]. Available: http://www.acmesystems.it/terra_modem_gprs. [Último acceso: Marzo 2016].
- [35] M. T. Costales Medina, «Análisis del Protocolo CAN (Controller Area Network) e Implementación de un Prototipo de Control de Nodos Interconectados por un Bus de Comunicaciones CAN,» de *Proyecto Final de Carrera*, Escuela Politécnica Nacional, Quito, 2008, p. 30.
- [36] Microchip, «Microchip MCP2515,» 2016. [En línea]. Available: <http://www.microchip.com/wwwproducts/en/MCP2515>. [Último acceso: 12 Julio 2016].
- [37] G. Avendaño, «Motor,» 7 Abril 2014. [En línea]. Available: http://www.motor.com.co/industria-en-marcha/ARTICULO-WEB-NEW_NOTA_INTERIOR-13796615.html. [Último acceso: 25 Octubre 2014].

ANEXOS

ANEXO 1 Reseña Histórica del Protocolo CAN Bus

El protocolo CAN surgió debido a la demanda de una mayor comodidad en los vehículos modernos, esto hizo que se incrementara los dispositivos electrónicos dentro del vehículo tales como ventanillas accionadas eléctricamente, ajustes de espejos, control de luces, aire acondicionado, bloqueo central, sistemas de administración de motor, entre otros, proporcionando así confort y seguridad al conductor. Por esta razón se hizo necesario desarrollar un sistema para intercambiar información entre los diferentes sistemas de control y sus sensores asociados.

Este intercambio de información anteriormente se realizaba mediante sistemas de cableado punto a punto, pero las necesidades fueron aumentando de tal forma que se necesitaban cuantiosos cables y muchos conectores, este incremento trajo consigo el aumento en el costo de los materiales, y el tiempo de producción del vehículo. De ahí nació la necesidad de buscar la solución a este problema introduciendo un sistema de bus serial de alta integridad para comunicar estos dispositivos inteligentes.

En el año 1983 Robert Bosch GmbH inicio el proyecto de diseño y desarrollo de un nuevo sistema de comunicaciones industriales que permita enlazar varios dispositivos electrónicos instalados en el automóvil, reduciendo el cableado y además con una muy buena funcionalidad, fue dirigido por Siegfried Dais, Martin Litschel y el Dr. UweKienche [13].

En febrero de 1986 se presentó oficialmente el sistema de bus serial CAN ante el congreso de la sociedad de ingenieros automotrices SAE (Society of Automotive Engineers) celebrado en Detroit – Estados Unidos. Originalmente fue desarrollado para aplicaciones en automotores, pero a través del tiempo su uso se ha extendido para otras aplicaciones como por ejemplo en sistemas de control embebidos para máquinas, dispositivos médicos, aplicaciones domésticas, entre otros.

En el año 1987 llegan los primeros controladores CAN al mercado, uno de los fabricantes fue la empresa INTEL con el primer chip llamado FullCAN-Chip 82526, después de un tiempo Philips Semiconductores también terminó el desarrollo de su primer chip, el 80C200. Estos chips constituyen la base para los controladores actuales, pero con una diferencia notable en cuanto al filtro de aceptación y manejo de errores. INTEL adoptó el concepto de FullCAN, ya que este requería

menos carga de la CPU del microcontrolador que la implementación BasicCAN elegida por Philips. Pero por otra parte, el dispositivo de FullCAN era limitado en cuanto al número de mensajes que podían ser recibidos, y además con respecto a la fabricación el BasicCAN requería menos silicio, lo que reducía su costo. Actualmente los términos BasicCAN y FullCAN han quedado obsoletos [13] y [23].

En 1991, Robert Bosch GmbH publica la especificación CAN en su versión 2.0; posteriormente dicha especificación fue sometida a su estandarización internacional [13], luego en 1992 Mercedes-Benz fue uno de los pioneros en implementar la red CAN Bus en sus automóviles de clase alta, seguido por grandes empresas automotoras como BMW, Renault, Volkswagen, entre otras. Pero en un comienzo este bus era limitado, ya que solamente interconectaban las unidades de control electrónico que se dedicaban a cuidar el motor, y con el pasar del tiempo fueron interconectando las demás unidades de los diferentes componentes electrónicos del automóvil. Por esta razón se implementan dos sistemas de CAN Bus separados físicamente, y conectados a través de “gateway” que es un equipo para interconectar redes.

En este mismo año usuarios internacionales y grupos de fabricantes fundaron oficialmente la organización CiA (CAN en la Automoción) promovida por Holger Zeltwanger, sin fines de lucro, con el objetivo de proporcionar información técnica de productos y comercialización para el uso del CAN Bus, La primera publicación técnica de CiA fue un artículo donde se especificaba la capa física del protocolo y también hacían la recomendación de uso de transceptores CAN que cumplieran con las normas de la organización internacional de normalización ISO (International Organization for Standardization) 11898. Posteriormente la CiA publica la especificación de la capa CAN denominada CAN capa de aplicación CAL (CAN Application Layer), que se desarrolló a partir del material existente de los sistemas médicos de Philips y de STZP [23].

Como se ve plasmado en la investigación [13], en Noviembre de 1993, el protocolo CAN es estandarizado bajo la norma ISO 11898 donde se define la capa física para velocidades de transferencia de datos de hasta 1 Mbps; un bus con tolerancia a fallos para las aplicaciones de baja velocidad hasta 125Kbit/s estandarizado bajo la norma ISO 11519 actualizado posteriormente por la ISO 11519-2, luego en 1994, se organizó la primera conferencia internacional acerca del CAN por parte de la CiA, y la firma Allen-Bradley publicó un protocolo de capa de aplicación CAN denominado Device Net. Posteriormente en 1995 se publica una mejora a la norma ISO 11898 en la que se añaden 29 bits como identificadores que hasta ese momento eran de 11 bits, además se incluyeron algunas recomendaciones para la especificación CAN 2.0B.

Por otra parte las especificaciones CAN con el tiempo han sido revisadas y estandarizadas en diferentes secciones como son: la norma ISO 118998-1 describe la capa de transmisión de datos CAN; la norma ISO 11898-2 la capa física CAN no tolerante a fallos; y la ISO 11898-3 la capa física CAN tolerante a fallos. Además de los estándares de ISO 11992 referente a la interfaz para camiones y remolques e ISO 11783 referente a la maquinaria agrícola y forestal, definen los perfiles del uso CAN.

Ya por último en el año 2000, ISO formó un grupo integrado por Demd Mueller y Tomas Fuehrer ingenieros de Bosch, y algunos académicos y expertos en la industria de los semiconductores, los cuales desarrollaron el nuevo protocolo de comunicaciones en tiempo real para CAN TTCAN (Time-Triggeredcommunication of CAN), este protocolo permitía la transmisión de mensajes simultáneos, como la implementación de un bucle cerrado para el control del bus; y como el protocolo del bus no fue modificado, este tipo de mensajes podían seguir siendo enviados con el mismo sistema físico del bus [23].

ANEXO 2 Descripción de Capas del Protocolo CAN Bus

1. CAPA DE ENLACE

La capa de enlace de datos del protocolo CAN define el control de acceso al medio y el control lógico para la transmisión de mensajes, de acuerdo al modelo de referencia OSI. Además esta capa se divide en dos subcapas, el control de enlace lógico LLC (Logical Link Control), y el control de acceso medio MAC (Medium Access Control).

1.1 CONTROL ENLACE LOGICO (LLC)

Esta subcapa es la encargada del filtrado de los mensajes, la notificación de sobrecarga, y la administración de la recuperación.

1.1.1 Filtrado de Mensajes

Dado que cada mensaje tiene su propio identificador, este no me especifica hacia quien va dirigida dicha trama, solamente define su contenido, entonces a través del filtrado los módulos receptores del sistema determinan si el mensaje deberá ser o no procesado.

1.1.2 Notificación de Sobrecarga

Cuando un módulo receptor requiera un atraso en la siguiente trama, la subcapa LLC genera una trama de sobrecarga, cabe destacar que como máximo se pueden generar dos tramas de sobrecarga.

1.1.3 Proceso de Recuperación

Durante el funcionamiento del sistema CAN se puede dar el caso que algunas de las tramas enviadas por los módulos al bus se pierdan, o también suele suceder que se pierda el arbitraje, generando así algunos errores durante la transmisión; este protocolo proporcionará la retransmisión automática de dichas tramas.

1.2 CONTROL DE ACCESO AL MEDIO

Es el núcleo del protocolo CAN que se encarga del encapsulamiento y desencapsulamiento de datos, arbitraje a la hora de acceder al bus; otra de las funciones es la detección de errores y su respectiva señalización, el aislamiento de fallos en unidades de control y la identificación del estado del bus para poder iniciar una nueva transmisión o recepción de un mensaje.

1.2.1 Acceso Múltiple y Arbitraje de Acceso al Medio

Una de las características que distingue al protocolo CAN es su técnica de acceso al medio denominada: Acceso múltiple con detección de portada, detección de colisión + resolución de colisión CSMA/CD+CR (Carriersense, Multiple Access/ Collision Detection + Collision Resolution). Cada nodo debe vigilar el bus en un periodo sin actividad antes de enviar un mensaje, además una vez que ocurra el periodo sin actividad cada nodo tiene la misma oportunidad para enviar un mensaje (multipleaccess), en caso de que dos nodos intenten transmitir al mismo tiempo se detectará la colisión [13].

El método de acceso al medio utilizado por el CAN Bus añade una característica muy importante que es la resolución de colisión, que se basa en una topología eléctrica que aplica una función lógica AND a cada bit, se resuelve dándole mayor prioridad a un bit de tipo dominante. Se define como bit dominante al equivalente de valor lógico "0", y bit recesivo al valor lógico "1". Cada transmisor escucha continuamente el valor presente en el bus, y se retira cuando su valor no coincide, en cambio cuando hay coincidencia la transmisión continua sobreviviendo el mensaje que tenga un identificador con mayor prioridad. Los demás nodos tendrán que esperar un tiempo para reintentar la transmisión.

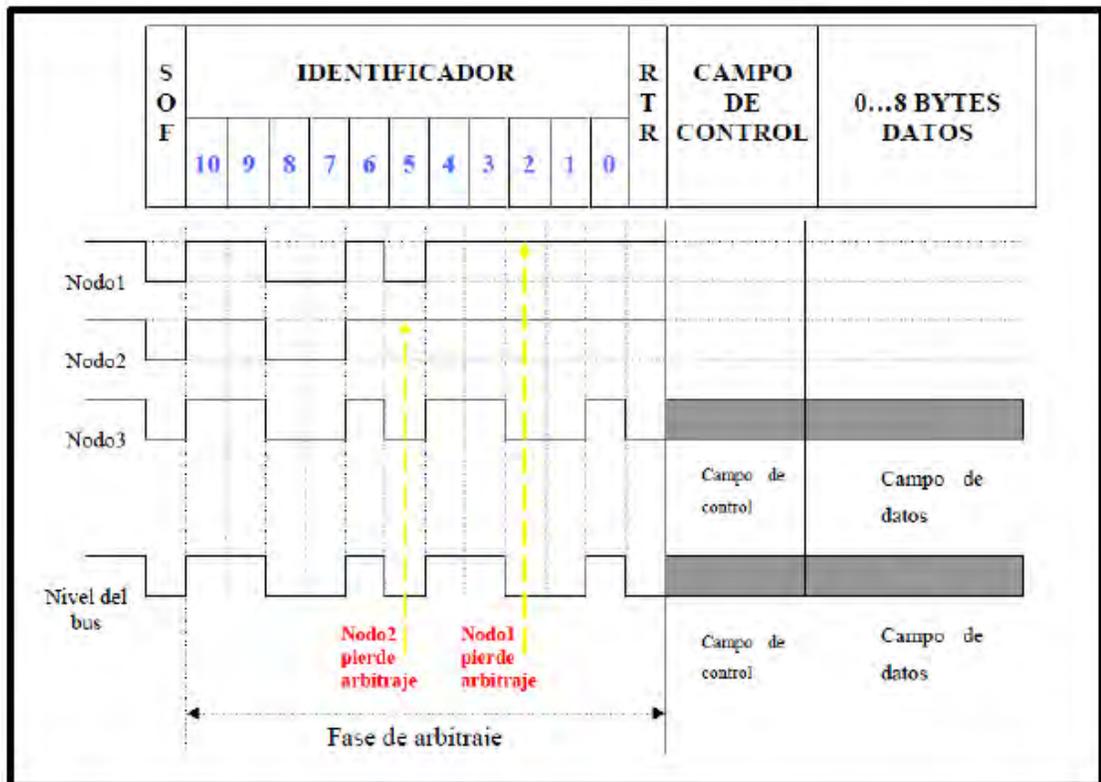
Cuando se detecta la colisión es ahí donde se comienza el arbitraje. Cada nodo que quiere enviar un mensaje, primero que todo envía los bits del identificador de la trama y vigila el bus. El nodo de mayor prioridad será el que tenga sus bits en estado dominante o sea un valor de "0", accederá primero al bus y una vez pasado un tiempo de espera diferente para cada nodo se vuelve a intentar el acceso sin repetición de mensaje ni pérdidas de tiempo. Los mensajes permanecen intactos a pesar de las colisiones, además que no se generará ningún retraso en los mensajes de alta prioridad.

En el CAN Bus se asigna un identificador de mensaje a un nodo en específico, con el fin de evitar que dos nodos puedan iniciar la transmisión simultánea de mensajes con el mismo identificador, pero con datos diferentes. El protocolo CAN establece que cada mensaje es único en el sistema, por ejemplo la variable "Velocidad", que debe ser transmitida por un nodo determinado, con un identificador concreto y con una longitud fija concreta y coherente con la codificación de la información en el campo de datos.

1.2.1.1 Secuencia de un Proceso de Arbitraje del CAN Bus entre Tres Nodos

Los nodos 1,2 y 3 inician el proceso de transmisión de tramas al mismo tiempo, durante la transmisión cada nodo emisor comprueba en cada bit dentro del campo de arbitraje si todavía está autorizado para transmitir. El nodo 2 pierde su acceso en el bit 5, lo mismo ocurre en el nodo 1 que pierde arbitraje en el bit 2. Bajo estas condiciones el bit de trama de petición de transmisión remota RTR (Remotetransmission Request Frame), define como dominante para la trama de datos al nodo 3 y recesivo para las otras dos tramas o sea los nodos 1 y 2 resolviendo así la colisión; consecuentemente el nodo 3 continuará con el proceso de transmisión, como muestra la Figura 56 [12].

Figura 56. Proceso de arbitraje para la transmisión de datos



Fuente: Estudio para Simular una Red CAN con aplicación en Comunicación de Dispositivos Electrónicos en el Automóvil [13]

2. CAPA FÍSICA

Se encarga de la transferencia de los bits entre los diferentes módulos que componen la red, especifica el medio físico por el cual se realizará la transmisión de los datos.

Cuando se generó la especificación original de CAN, la capa física no fue definida pero a través del tiempo y con el estándar ISO 11898 se establecieron las características de las señales eléctricas para aplicaciones de alta velocidad y con el estándar ISO 11519 para aplicaciones de baja velocidad.

2.1 MEDIO FÍSICO

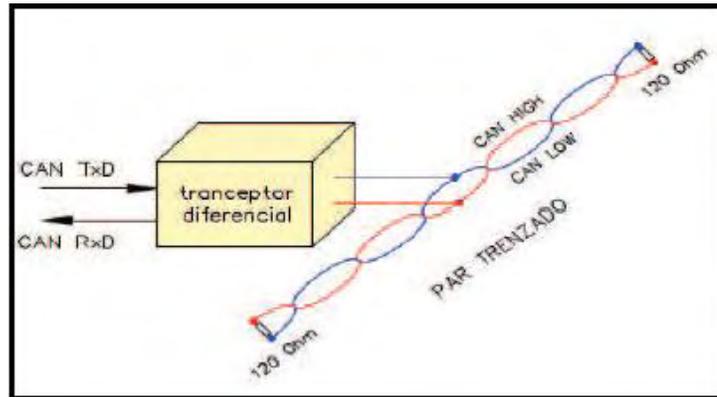
Consiste en un cable de par trenzado UTP (Unshielded Twisted Pair) de acuerdo a la normativa ISO 11898, con transmisión diferencial. El trenzado de los hilos sirve para anular los campos magnéticos, por lo cual es recomendable no modificar la longitud de estos. Uno de los cables se llama CAN_H (High) en donde el valor de tensión oscila entre 2,75V a 5V, y el otro es CAN_L (Low) en donde el valor de tensión oscila entre los 0V a 2,25V.

A los extremos de estos cables, van conectados los elementos de cierre que son dos resistencias, una a cada lado, como se puede apreciar en la Figura 57, su valor habitual es de 120 ohmios. Las resistencias sirven para impedir posibles efectos parásitos que puedan perturbar la información transmitida, haciendo que los mensajes lleguen con algunos errores a cada módulo, por otra parte, las resistencias sirven para representar la impedancia de las líneas.

Una de las ventajas que tiene el CAN Bus es que si se interrumpe la línea CAN_H o que se derive a masa, el sistema trabajará con la línea CAN_L con respecto a masa, y en caso de que se interrumpa la línea CAN_L ocurrirá lo contrario. Esto es muy importante ya que permite que el sistema trabaje normalmente con alguno de los cables cortados o conectados a masa, la única forma que hace que el sistema quede totalmente sin funcionamiento es cuando se cortan los dos cables.

Este arreglo tiene una gran ventaja, ya que podemos utilizar los dispositivos puente a los repetidores para aumentar el número de módulos del bus que pueden ser conectados. La Figura 57 muestra el medio físico del CAN Bus.

Figura 57. Medio físico del CAN Bus



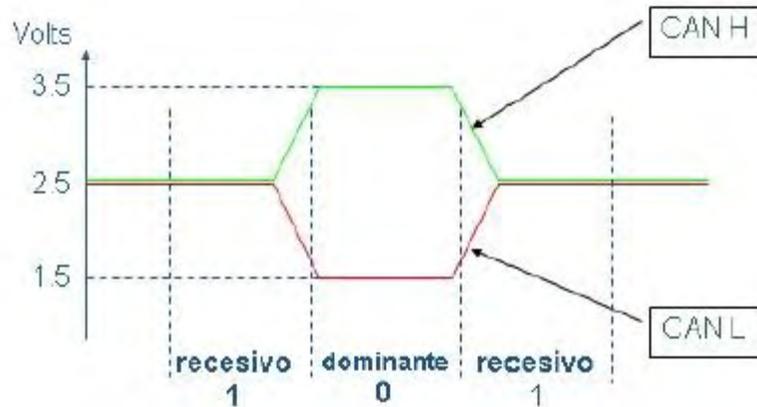
Fuente: Diseño e Implementación del Protocolo CAN para el control de un Módulo de Red de Sensores [14]

2.2 NIVELES ELECTRÓNICOS DE TRANSMISIÓN

Todos los módulos conectados al bus interpretan dos niveles lógicos [13].

- **Dominante:** La tensión diferencial (CAN_H-CAN_L) es del orden 2.0V Con CAN_H=3.5V y CAN_L=1.5V (nominales); equivale al nivel lógico cero.
- **Recesivo:** La tensión diferencial (CAN_H-CAN_L) es del orden 0V Con CAN_H=CAN_L= 2.5V (nominales); equivale a nivel lógico uno.

Figura 58. Niveles de tensión, (estándar ISO11898)



Fuente: Estudio para Simular una Red CAN con aplicación en Comunicación de Dispositivos Electrónicos en el Automóvil [13]

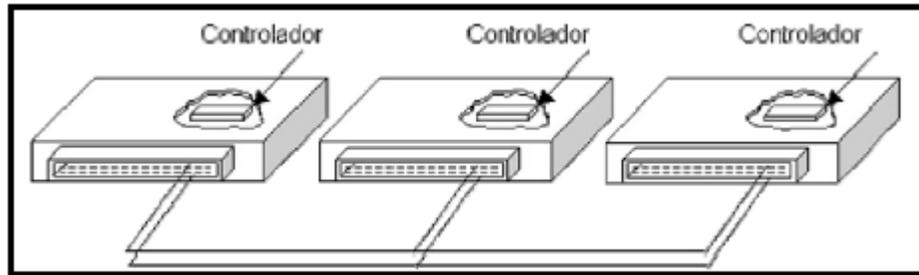
2.3 ELEMENTOS DEL SISTEMA CAN BUS

Una red CAN Bus para automoción está constituida por un controlador y un transceptor ubicados dentro de las unidades de control y dos cables para la transmisión de datos (Bus) [24].

2.3.1 Controlador

Es el encargado de la conexión entre el microprocesador del módulo y el transceptor. Su función es acondicionar la información que entra y sale de cada uno de los componentes. Se encuentra ubicado en el interior de cada unidad de control como muestra la Figura 59, trabaja con niveles de tensión muy bajos en el orden de milivoltios, y es el que determina la velocidad de transmisión del mensaje. Los controladores también intervienen en la sincronización de los módulos para realizar una correcta emisión y recepción de mensajes.

Figura 59. Controladores de los componentes del CAN Bus



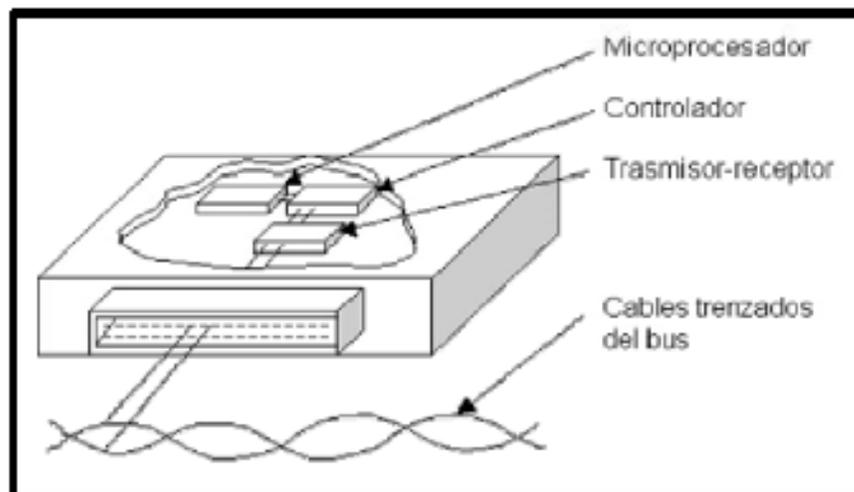
Fuente: Estudio para Simular una Red CAN con aplicación en Comunicación de Dispositivos Electrónicos en el Automóvil

2.3.2 Transceptor (Transmisor - Receptor)

El transceptor tiene como función recibir y transmitir los datos, además es un elemento acondicionador de señales de tensión, para esto emplea amplificadores de señal que sitúan los niveles de tensión de forma adecuada cuando la información se trasmite a las líneas del bus, y la reduce cuando es recogida para poderla suministrar al controlador.

El transceptor es un circuito integrado que está situado en cada uno de los módulos conectados a la red, se puede observar en la Figura 60, trabaja con una corriente aproximada de 500mA, y nunca interfiere en la modificación del contenido del mensaje. La comunicación de este, sólo puede ser semiduplex, lo que significa que pueden enviarse señales entre dos terminales en ambos sentidos pero no simultáneamente, es decir si un nodo está transmitiendo el otro solamente puede recibir la información a través de su transceptor.

Figura 60. Transceptores de red CAN Bus



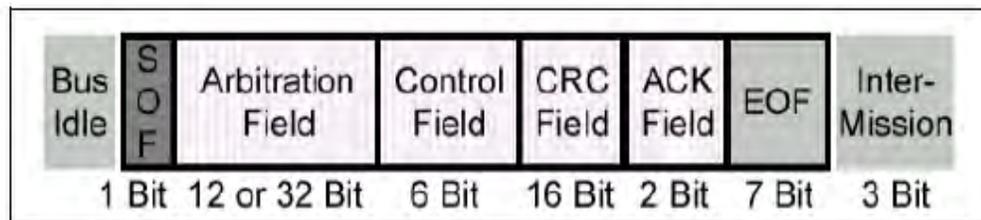
Fuente: Estudio para Simular una Red CAN con aplicación en Comunicación de Dispositivos Electrónicos en el Automóvil

ANEXO 3 Tipos de Tramas

1. TRAMA REMOTA

Los nodos tienen habilidad para solicitar información a otros nodos. Un nodo pide una información a los otros y el nodo que tiene dicha información envía una comunicación con la respuesta que puede ser recibida por otros nodos si están interesados. El mensaje de transmisión remota se muestra en la Figura 61.

Figura 61. Mensaje de transmisión remota



Fuente: BUS CAN [15]

En este tipo de mensaje se envía una trama con el identificador de nodo requerido. A diferencia con los mensajes de datos, el bit RTR toma un valor recesivo y no hay campo de datos. En caso de que se envíe un mensaje de datos y de petición remota con el mismo identificador, el de datos ganará el acceso al bus debido a que este tiene un valor de RTR dominante.

2. TRAMA DE ERROR

Esta trama es generada por cualquier nodo que detecte algún error durante la transmisión o recepción de una trama de datos o remota; está formada por dos campos. Uno de éstos es el indicador de error y el otro es el delimitador de error. El delimitador de error consta de 8 bits recesivos consecutivos y permite reiniciar a los nodos la comunicación sin ningún problema tras el error. En cambio el indicador es distinto, porque se basa en el estado de error del nodo que detecta el error [15].

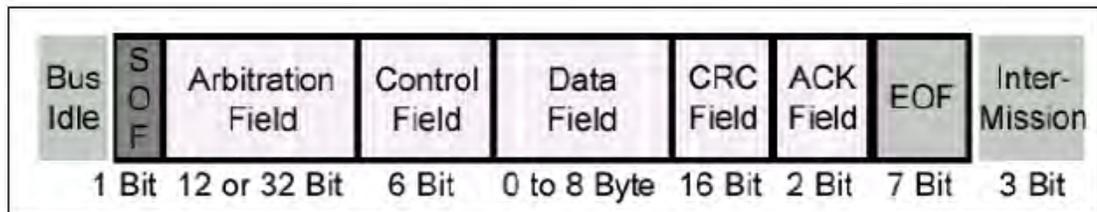
- Un nodo en estado de error “Activo” detecta un error en el bus, Y se interrumpe la comunicación del mensaje generando un “Indicador de error activo” que consiste en una secuencia de 6 bits dominantes sucesivos. Esta

secuencia rompe la regla de relleno de bits generando tramas de error en los otros nodos. Por tanto el indicador de error puede extenderse entre 6 y 12 bits dominantes sucesivos. Finalmente, se recibe el campo de delimitación de error que es formado por 8 bits recesivos, lo cual hace que la comunicación se reinicie y el nodo que había sido interrumpido reintenta hacer la transmisión del mensaje.

- El otro caso es en estado de error “Pasivo” cuando se detecta un error, el nodo transmite un “Indicador de error pasivo” seguido por el campo delimitador de error. El indicador de error de tipo pasivo consiste en 6 bits recesivos seguidos, la trama de error para un nodo pasivo es una secuencia de 14 bits recesivos. Por esta razón una trama de tipo pasivo no afectará a ningún nodo en la red, excepto cuando el error es detectado por el nodo que está transmitiendo, esto hará que los demás nodos detecten una violación de las reglas de relleno por lo tanto transmitirán tramas de error.

Otro método para la detección de errores es a través del análisis de la trama que se muestra en la Figura 62.

Figura 62. Trama de error



Fuente: BUS CAN [15]

- **Campo de chequeo de errores CRC:** este campo contiene información adicional a la trama, se calcula con un polinomio generador en el receptor y en el emisor. Permite detectar errores aleatorios hasta de 5 bits o una secuencia de 15 bits corruptos.
- **Campo de reconocimiento ACK:** para el caso del emisor será recesivo y para el receptor este deberá sobrescribirlo como dominante y el primero lo comprobará, mediante la monitorización que el mensaje ha sido escuchado. Si no sucede así, la trama se considerará corrupta.

3. TRAMA DE SOBRECARGA

Es utilizada para que un nodo receptor retrase la transmisión de la trama siguiente, durante el espacio entre tramas. Esto hace que se diferencie ante una trama de error, debido a que tienen el mismo formato. La trama de sobrecarga consta de dos campos: el indicador de sobrecarga y el delimitador. El indicador de sobrecargar consta de 6 bits dominantes que pueden ser seguidos por los generados por otros módulos, dando lugar a un máximo de 12 bits dominantes. El delimitador es de 8 bits recesivos.

Una trama de sobrecarga puede ser generada por cualquier nodo, que debido a sus condiciones internas, no está en condiciones de iniciar la recepción de un nuevo mensaje. De esta forma retrasa el inicio de transmisión de un nuevo mensaje. Un nodo puede generar como máximo 2 tramas de sobrecarga consecutivas para retrasar un mensaje. Otra razón para iniciar la transmisión de una trama de sobrecarga es la detección por cualquier nodo de un bit dominante en los 3 bits de "inter-trama". Por todo ello una trama de sobrecarga generada por un nodo dará normalmente lugar a la generación de tramas de sobrecarga por los demás nodos dando lugar, como se ha indicado, a un máximo de 12 bits dominantes de indicador de sobrecarga [15].

4. ESPACIO ENTRE TRAMAS

El espacio entre tramas separa a una trama de cualquier tipo de la siguiente trama que puede ser de datos o interrogación remota. El espacio entre tramas consta de al menos 3 bits recesivos. Una vez que transcurre esta secuencia un nodo en estado de error activo puede iniciar nuevamente una transmisión o también puede suceder que el bus permanezca en reposo. Otro caso es para un nodo en estado de error pasivo, que deberá esperar una secuencia adicional de 8 bits recesivos antes de poder iniciar una transmisión. De esta manera se asegura una ventaja en inicio de transmisión a los nodos en estado activo frente a los nodos en estado pasivo [23].

ANEXO 4 Pines del Microcontrolador ATmega328P-PU

ATmega328P-PU

	(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)	A5
D0	(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)	A4
D1	(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)	A3
D2	(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)	A2
D3	(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)	A1
D4	(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)	A0
	VCC	7	22	GND	
	GND	8	21	AREF	
	(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC	
	(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)	D13
D5	(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)	D12
D6	(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)	D11
D7	(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)	D10
D8	(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)	D9

Fuente: Datasheet ATmega328P-PU

URL: http://www.atmel.com/images/atmel-8271-8-bit-avr-microcontroller-atmega48a-48pa-88a-88pa-168a-168pa-328-328p_datasheet_summary.pdf