



1859

UNIVERSIDAD NACIONAL DE LOJA

FACULTAD DE LA ENERGÍA LAS INDUSTRIAS Y LOS
RECURSOS NATURALES NO RENOVABLES

INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES

TÍTULO

“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA
ELECTRÓNICO PARA LA MEDICIÓN Y REGISTRO DE
VELOCIDAD DE UN VEHÍCULO MEDIANTE GPS”

TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

Autor: Roberto Gregorio Zambrano Vera

Director: Ing. Andy Fabricio Vega León, Mg. Sc.

Loja - Ecuador

2019


CERTIFICACIÓN

Ing. Andy Fabricio Vega León, Mg. Sc.
DIRECTOR DEL TRABAJO DE TESIS

CERTIFICA:

Haber dirigido, asesorado, revisado y corregido el presente trabajo de tesis de grado, en su proceso de investigación cuyo tema versa en **“Diseño e Implementación de un Sistema Electrónico para la Medición y Registro de Velocidad de un Vehículo Mediante GPS”** previo a la obtención del título de **Ingeniero en Electrónica y Telecomunicaciones**, realizado por el egresado: **Roberto Gregorio Zambrano Vera**, mismo que cumple con la reglamentación y políticas de investigación, por lo que autorizo su presentación y posterior sustentación y defensa.

Loja, 01 de agosto de 2019

A handwritten signature in blue ink, appearing to read 'Andy Fabricio Vega León', is written over a horizontal line. Below the signature is a dotted line.

Ing. Andy Fabricio Vega León, Mg. Sc.
DIRECTOR DEL TRABAJO DE TESIS

AUTORÍA

Yo, **ROBERTO GREGORIO ZAMBRANO VERA**, declaro ser autor del presente trabajo de tesis y eximo expresamente a la Universidad de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de la misma.

Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi trabajo de tesis en el Repositorio Institucional- Biblioteca Virtual.

Firma:



Cédula: 0924499593

Fecha: 29/11/2019

CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL AUTOR, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.

Yo, **ROBERTO GREGORIO ZAMBRANO VERA**, declaro ser autor de la tesis titulada **“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA ELECTRÓNICO PARA LA MEDICIÓN Y REGISTRO DE VELOCIDAD DE UN VEHÍCULO MEDIANTE GPS”**, como requisito para optar al grado de: **INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES**; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los dos días del mes de Julio del dos mil diecinueve.

Firma: .....

Autor: Roberto Gregorio Zambrano Vera

Cédula: 0924499593

Dirección: Loja (Bolívar y Catacocha)

Correo electrónico: robertofish@hotmail.es – rgzambranov@unl.edu.ec

Teléfono: 072580954 **Celular:** +593 968544955

DATOS COMPLEMENTARIOS

Director de tesis: Ing. Andy Fabricio Vega León, Mg. Sc.

Tribunal de grado: Ing. John Jossimar Tucker Yépez, Mg. Sc.

Ing. Pabel Rodolfo Merino Vivanco, Mg. Sc.

Ing. Cristian Hernán Campoverde Ramírez, Mg. Sc.

DEDICATORIA

Dedicado a Maidelline Zambrano

AGRADECIMIENTO

Agradezco principalmente a Dios que Siempre estuvo conmigo en todo momento, su confort me ayudo en tiempos difíciles.

A mis padres quienes a pesar de pasar por situaciones difíciles supieron ayudarme y guiarme con sus sabios consejos.

A mis docentes que siempre tuvieron paciencia y supieron impartir todos los conocimientos a lo largo de mi vida universitaria

A mi director de tesis Ing. Andy Vega por su preocupación, dedicación y por guiarme durante el desarrollo de este trabajo

También agradezco a mis compañeros por todos los momentos de amistad y de fraternidad dentro y fuera del aula y a la Universidad Nacional de Loja por brindarme la oportunidad de cursas por sus aulas

TABLA DE CONTENIDO

CERTIFICACIÓN.....	II
AUTORÍA	III
CARTA DE AUTORIZACIÓN	IV
DEDICATORIA.....	V
AGRADECIMIENTO.....	VI
TABLA DE CONTENIDO.....	VII
ÍNDICE DE FIGURAS.....	XI
ÍNDICE DE TABLAS.....	XIV
1. TÍTULO.....	1
2. RESUMEN	2
2.1 ABSTRACT.....	3
3. INTRODUCCIÓN.....	4
4. REVISIÓN DE LA LITERATURA.....	5
4.1 SISTEMA ACTUAL EN EL PAÍS Y EN LA CIUDAD DE LOJA.....	5
4.1.1 Sistemas para fотomultas en el país.....	5
4.1.2 Sistemas de fотomultas en la ciudad de Loja	6
4.1.2.1 Cámaras fijas	6
4.1.2.2 Cámaras móviles	7
4.1.2.3 Cartel informativo	7
4.2 FUNCIONAMIENTO DE SISTEMA DE FOTOMULTAS EN LA CIUDAD DE LOJA.....	8
4.2.1 Funcionamiento básico del sistema de fотomultas en la ciudad Loja y notificación a infractores.....	8
4.2.1.2 Funcionamiento del sistema de fотomultas mediante cámaras fijas.....	8
4.2.1.2 Funcionamiento de los carteles informativos	10
4.2.1.3 Notificación a los posibles infractores.....	13
4.3 FUNCIONAMIENTO DEL SISTEMA DE POSICIONAMIENTO GLOBAL (GPS).	14
4.3.1 Composición del sistema de GPS.....	14
4.3.2 Principio matemático del GPS	15
4.3.3 Formatos GPS	17
4.3.3.2 Estándar NMEA-2000	17
4.4 GSM Y GPRS	18

4.4.1 Red para el Sistema global para las comunicaciones móviles o Global System for Mobile Communications (GSM)	18
4.4.2 Servicio general por paquetes vía radio o general Packet Radio Service (GPRS)	18
4.4.3 Ventajas de GPRS	18
4.5 BASE DE DATOS.....	19
4.5.1 Concepto de base de datos	19
4.5.2 Tipo de base de datos.	19
4.5.3 Base de datos en forma de texto.....	19
5. MATERIALES Y MÉTODOS.....	21
5.1 Metodología.....	21
5.2 DISEÑO DEL SISTEMA PARA LA OBTENCIÓN DE POSICIÓN, HORA Y VELOCIDAD.	21
5.2.1 Desarrollo cronológico del sistema para la obtención de posición, hora, velocidad y envío de datos.	21
5.3 SELECCIÓN DE LOS COMPONENTES	22
5.3.1 Selección del Microcontrolador	22
5.3.1.1 Arduino uno.....	23
5.3.1.2 Desventajas del Arduino Uno:.....	24
5.3.1.3 Ventajas de Arduino Uno:	24
5.3.1.4 Microchip familia 16FXX	24
5.3.1.5 PIC 16F628A.....	24
5.3.1.6 Características del PIC 16F628A	25
5.3.1.7 Desventajas del PIC 16F628A.....	25
5.3.1.8 Ventajas del PIC 16F628A	25
5.3.1.9 PIC 18F4620.....	25
5.3.1.10 características.....	26
5.3.1.11 Características adicionales.....	26
5.3.1.12 Ventajas del PIC 18F4620.....	26
5.3.1.13 Desventajas del PIC 18F4620.....	27
5.3.1.14 Microcontrolador elegido	27
5.3.2 Selección del dispositivo GPS	29
5.3.2.1 Receptor GPS P28146 DE Parallax.....	29
5.3.2.2 Características técnicas [26]	29
5.3.2.3 Desventajas del GPS P28146	29

5.3.2.4	Ventajas del GPS P28146.....	30
5.2.2.5	Modulo GPS u-blox 6.....	30
5.2.2.6	Características del módulo u-blox 6 GPS [27]	30
5.3.2.7	Ventajas del módulo GPS u-blox 6	31
5.3.2.8	Desventajas del módulo GPS u-blox 6.....	31
5.3.2.9	modulo GPS elegido.....	31
5.3.4	SELECCIÓN DEL MÓDULO GPRS.....	32
5.3.4.1	Modulo GSM/GPRS SIM 900.....	33
5.3.4.2	características del módulo GPS/GPRS SIM900 [28]	33
5.3.4.3	Ventajas del GSM/GPRS SIM 900	33
5.2.3.4	Desventajas del GSM/GPRS SIM 900	34
5.3.4.5	Modulo GSM/GPRS SIM 800L	34
5.3.4.6	Características del módulo GSM/GPRS SIM 800L [28]	34
5.3.4.7	Ventajas del módulo GSM/GPRS SIM 800L [29].....	34
5.3.4.8	Desventajas del módulo GSM/GPRS SIM 800L	35
5.3.4.9	Modulo GSM/GPRS elegido.....	35
5.3.5	Modulo micro SD.....	36
5.2.4.1	Características del módulo micro SD Catalex	36
5.4	MATERIALES.....	36
5.5	Diseño del sistema electrónico para la medición y registro de velocidad	37
5.6	Simulación del circuito	39
5.6.1	Programas usados para la simulación.....	39
5.6.2	Desarrollo de la simulación del circuito para la medición y registro de velocidad	40
5.6.3	Integración del módulo GPS a la simulación.....	41
5.6.4	Configuración de la SD Card y ejecución de la simulación.....	42
5.7	Implementación del código en el microcontrolador	51
5.8	código para visualización de los datos obtenidos.....	53
5.8.1	Software para la elaboración del código para la visualización de los datos obtenidos.....	53
5.8.2	Desarrollo del código para la visualización y almacenamiento de los datos obtenidos	55
5.8.3	Desarrollo de plataforma de posicionamiento.....	58
6.	RESULTADOS	60

6.1 Resultados en el PCB para pruebas	60
6.2 Resultados del PCB final	63
6.3 Resultados obtenidas en un vehículo.	67
6.4 Visualización de los datos obtenidos a través de la PC	67
6.5 Visualización mediante plataforma de posicionamiento	69
7. DISCUSIÓN.....	74
8. CONCLUSIONES.....	75
9. RECOMENDACIONES	76
10.BIBLIOGRAFÍA.....	77
11.ANEXOS	80
ANEXO 1: CÓDIGO EN MIKROC PARA PIC 18F4620	80
ANEXO 2: Código Labview complete.....	92
ANEXO 3: Hojas de datos (SIM800L, GPS U-blox 6, PIC18F4620)	95
ANEXO 4: Solicitud de autorización dirigida a la ANT (Agencia Nacional de Transito).....	99

ÍNDICE DE FIGURAS

Figura 1. Cámara Speedcurb. [6].....	6
Figura 2. Camara por radar movil ARH S1. [6]	7
Figura 3. Indicador de velocidad. [8]	7
Figura 4. El sistema detecta tres fotografías del vehículo infractor. [9].....	8
Figura 5. Sensores Piezoeléctricos en asfalto. Fuente: El autor.	10
Figura 6. Ubicación correcta del cartel informativo. [12]	13
Figura 7. Ubicación con ángulo del cartel informativo. [12]	13
Figura 8. Estaciones terrenas para control de GPS en todo el mundo. [16]	15
Figura 9. Forma grafíaca de la trilateración para obtener un punto determinado en la tierra. [17].....	15
Figura 10. Diagrama de bloques del Sistema Electrónico para la Medición y Registro de Velocidad. Fuente: El autor.	21
Figura 11. Módulo arduino UNO. [23].....	23
Figura 12. Microcontrolador PIC16F628A. [24]	24
Figura 13. Microcontrolador PIC16F628A. Fuente: El autor	25
Figura 14. Microcontrolador PIC18FF4620. [25]	28
Figura 15. Modulo GPS P28146. [26].....	29
Figura 16. Modulo GPS u-blox 6. [27].....	30
Figura 17. Modulo GPRS SIM 900. [28]	33
Figura 18. Modulo GSM/GPRS SIM800L. [29]	34
Figura 19. Modulo micro SD CATALEX. [30]	36
Figura 20. Funcionamiento del sistema para la medición y registro de velocidad. Fuente: El autor	38
Figura 21. Circuito para la medición y registro de velocidad. Fuente: El autor.....	39
Figura 22. PIC18F4620 y LED indicadores. Fuente: El autor.	41
Figura 23. Modulo micro SD dentro del circuito. Fuente: El autor.	42
Figura 24. Se crea un nuevo documento. Fuente: El autor.....	43
Figura 25. Selección de formato avanzado. Fuente: El autor.....	43
Figura 26. Se selecciona el formato FAT32. Fuente: El autor	44
Figura 27. Se carga los archivos en edit properties. Fuente: El autor	44
Figura 28. Mensaje de presentación obtenido desde la simulación en proteus 7.8. Fuente: El autor.	45
Figura 29. El dispositivo muestra el mensaje "Esperando conexión..." para dar tiempo que el módulo GPS establezca conexión. Fuente: El autor.	45
Figura 30. Si la micro SD se encuentra insertada correctamente se muestra "Inicio OK" y enciende el puerto B7. Fuente: El autor.	46
Figura 31. Si la micro SD no está insertada correctamente se muestra el mensaje "inicio fallido" y se enciende el led B4. fuente: El autor.	46
Figura 32. El código crea la carpeta DIR_A en la que se guardaran los datos adquiridos. Fuente: El autor.	47

Figura 33. El GPS recibe los datos de los satélites y pueden visualizarse en la pantalla LCD. Fuente: El autor.	47
Figura 34. Valores de latitud y longitud durante la simulación. Fuente: El autor.	48
Figura 35. Tramas del módulo GPS simulado en proteus obtenidas desde el Virtual Terminal. Fuente: El autor.	48
Figura 36. El código indica que los archivos han sido creados correctamente. Fuente: El autor.	49
Figura 37. El código indica por medio de la pantalla LCD que los datos se están escribiendo en la micro SD. Fuente: El autor.	49
Figura 38. El mensaje "texto escrito correcto" indica que los datos obtenidos por el módulo GPS han sido almacenados en la micro SD. Fuente: El autor.	50
Figura 39. Archivo creado en WinImage. Fuente: El autor.	50
Figura 40. Datos obtenidos desde la simulación en Proteus. Fuente: El autor.	51
Figura 41. Diagrama de flujo del código para el microcontrolador. Fuente: El autor.	52
Figura 42. Bloque para acceder al puerto serial. Fuente: El autor.	55
Figura 43. Lectura y escritura del puerto serial. Fuente: El autor.	55
Figura 44. Bloque encargado de cerrar el puerto serial. Fuente: El autor.	56
Figura 45. bloque encargado de crear directorio y guardar datos. Fuente: El autor.	56
Figura 46. Filtrado de String y conversión a dato numérico. Fuente: El autor.	57
Figura 47. panel de control del VI generado. Fuente: El autor.	57
Figura 48. Código para la plataforma de posicionamiento. Fuente: El autor.	58
Figura 49. PCB para pruebas. Fuente: El autor.	60
Figura 50. mensaje de presentación. Fuente: El autor.	61
Figura 51. mensaje de espera en el que se verifica el estado de la micro SD Card. Fuente: El autor.	61
Figura 52. Este mensaje se muestra si se desconecta el módulo GPS o se supera el tiempo de espera para establecer conexión GPS/satélites. Fuente: El autor.	61
Figura 53. Coordenadas obtenidas por el GPS. Fuente: El autor.	61
Figura 54. Velocidad obtenida por el módulo GPS con un vehículo estático. Fuente: El autor.	62
Figura 55. datos de fecha y hora obtenidos por el módulo GPS. Fuente: El autor.	62
Figura 56. Se presenta el mensaje "Archivos creados" que indica que la carpeta en la que se guardarán los archivos ha sido creada y el GPS está en conexión. Fuente: El autor.	62
Figura 57. mientras el sistema escribe los archivos en la micro SD Card se muestra "escribiendo". Fuente: El autor.	62
Figura 58. "Texto escrito correcto" indica q todo el proceso de almacenamiento se ha desarrollado con normalidad. Fuente: El autor.	63
Figura 59. PCB para el procesamiento de datos obtenidos desde GPS u blox-6. Fuente: El autor.	63
Figura 60. Sistema completo para la medición y registro de velocidad mediante GPS. Fuente: El autor.	64
Figura 61 pantalla LCD 16X2 vista desde tu parte adversa. Fuente: El autor.	65

Figura 62. Sistema para la medición y registro de velocidad completo dentro de su caja metálica. Fuente: El autor.....	65
Figura 63 Sistema para medición y registro de velocidad mediante GPS visto desde su parte frontal. Fuente: El autor.....	66
Figura 64. Conjunto de sub circuitos que conforman el receptor para visualizar los datos del GPS en la PC. Fuente. El autor.....	67
Figura 65. Datos almacenados en forma de texto obtenidos desde el GPS con un vehículo estático. Fuente: El autor.	68
Figura 66. Datos obtenidos por un vehículo en movimiento. Fuente: El autor.....	68
Figura 67 Datos almacenados en la micro SD Card con vehículo estático. Fuente: El autor.....	69
Figura 68 Datos de ubicación con vehículo estático. Fuente: El autor.....	69
Figura 69. vista principal de la plataforma, en el estado 0. Fuente: El autor.	70
Figura 70. Punto graficado obtenido Av. Manuel Agustín Aguirre. Fuente: El autor. .	71
Figura 71 Coordenadas obtenidas desde la plataforma. Fuente: El autor.	71
Figura 72. Imagen obtenida desde la aplicación de Google Maps. Fuente: El autor.	72
Figura 73. Datos de posición, fecha, hora y velocidad obtenidos desde el SIM800L. Fuente: El autor.	73
Figura 74. Posición graficada en Google Maps. Fuente: El autor.....	73
Figura 75. Código completo Labview (primera parte). Fuente: El autor.	92
Figura 76. Código completo Labview (segunda parte). Fuente: El autor.....	93
Figura 77-. Código completo Labview (tercera parte). Fuente: El autor.	94

ÍNDICE DE TABLAS

Tabla 1. Fotorradars homologados. [4].....	5
Tabla 2 Ecuaciones y parámetros para la obtención de imágenes para Speedcurb. [10] .	9
Tabla 3. Límites para vehículos livianos. [13]	11
Tabla 4. Límites de velocidad para transporte público. [13]	11
Tabla 5. Límites de velocidad para transporte de carga. [13].....	12
Tabla 6. Comparación de los diferentes microcontroladores estudiados para este proyecto. Fuente: El autor.	27
Tabla 7. Comparación de las propiedades entre GPS P28146 Y GPS u-blox 6. Fuente: El autor.	32
Tabla 8. comparación entre SIM900L Y SIM800L. Fuente: El autor.....	35
Tabla 6. Materiales empleados para la elaboración del circuito para la medición y registro de velocidad. Fuente: El autor.	37
Tabla 10. Comparación de las principales características de Python, Matlab y Labview. Fuente: El autor.	54

1.TÍTULO

“Diseño e Implementación de un Sistema Electrónico para la Medición y Registro de Velocidad de un Vehículo Mediante GPS”

2. RESUMEN

El siguiente trabajo se enfoca en el diseño e implementación de un dispositivo; usando como tecnología principal el Sistema de posicionamiento Global (GPS, Global Positioning System) para dar información sobre la velocidad, posición y fecha, adquirida durante el recorrido de un vehículo, para luego enviar esta información a una base de datos mediante la Red para el Sistema Global para las Comunicaciones móviles (GSM, Global System for Mobile Communications), esta información podría ser usada como evidencia ante una posible multa que se haya levantado por error.

Para el desarrollo de este dispositivo se usó un controlador de interfaz periférico, (PIC, Peripheral Interface Controller), un módulo GPS y un módulo GSM/GPRS. Su funcionamiento se basa en la triangulación y medición de la señal de retardo que se produce al realizar la comunicación entre el receptor y el satélite GPS

Además, el sistema genera una base de datos en la que se guarda la información de velocidad, posición y hora. Cada cinco segundos la información se actualiza y se envía por medio de un módulo GSM/GPRS al final del recorrido. El objetivo es usar esa información como evidencia para apelar una posible fotomulta o usar la información para actividades particulares. También el sistema busca una integración futura a un sistema más amplio que permita compartir esta información a una base de datos que contenga datos de varios vehículos registrados.

2.1 ABSTRACT

The following work focuses on the design and implementation of a device; using as a main technology the Global Positioning System (GPS, Global Positioning System) to give information on the speed, position and date, acquired during the journey of a vehicle, and then send this information to a database through the Global System for Mobile Communications (GSM, Global System for Mobile Communications), this information could be used as evidence for a possible fine that has been raised in error.

For the development of this device, a peripheral interface controller (PIC, Peripheral Interface Controller), a GPS module and a GSM/GPRS module were used. Its operation is based on the triangulation and measurement of the delay signal that occurs when communicating between the receiver and the GPS satellite

In addition, the system generates a database in which the speed, position and time information is saved. Every five seconds the information is updated and sent by means of a GSM/ GPRS module at the end of the tour. The objective is to use that information as evidence to appeal a possible photomult or use the information for particular activities. The system also seeks future integration into a broader system that allows sharing this information to a database that contains data from several registered vehicles.

3. INTRODUCCIÓN

El exceso de velocidad es una de las principales causas de accidentes en el país según la Agencia Nacional de Tránsito (ANT) ocupando el segundo lugar con 14.8% por debajo de la principal causa, que, según esta entidad, se la atribuye a conducir desatento a las condiciones de tránsito [1]. Por ello se han tomado acciones dentro de cada ciudad para reducir el índice de accidentes y para esto el control de velocidad es primordial.

En octubre del 2015, el municipio de Loja inicio el control de velocidad a través de fotorradars fijos y móviles mediante una resolución impuesta desde la alcaldía en la cual se establecían los límites de velocidad en la ciudad y las multas para los infractores. En enero y febrero del 2019 se generaron alrededor de 500 fotomultas en la ciudad de Loja [2], Y más de 15000 durante los años 2017 y 2018.

Si bien la tecnología para estas fotomultas en la ciudad de Loja se basa en sensores y en radares, algunas multas han sido levantadas por error, ya sea por mala manipulación de los equipos, deterioro, o algún desperfecto en la configuración y la notificación debe realizarse dentro de las 72 horas posteriores a la infracción, caso contrario queda sin validez dicha acción.

Muchos conductores han sido notificados aun cuando su velocidad estaba por debajo del límite establecido en la ordenanza la misma que establece que se puede realizar una impugnación para esta sanción según su artículo 179 de la Ley Orgánica de Transporte Terrestre.

Este proyecto busca realizar un dispositivo que se colocara en el interior de un vehículo que permita llevar un registro de velocidad a lo largo de un recorrido usando tecnología de GPS, esta información puede ser usada para la impugnación de una fotomulta; además los datos se enviara mediante GPRS hasta una base de datos buscando a priori un sistema de interconexión de vehículos donde se tenga un registro claro de velocidad, ubicación y hora, lo que permitirá un enfoque con menos errores y así reducir los accidentes por exceso de velocidad. El proyecto no busca remplazar el sistema de fotomultas si no trabajar de manera conjunta para tener varias fuentes al momento de levantar una infracción.

4. REVISIÓN DE LA LITERATURA

4.1 SISTEMA ACTUAL EN EL PAÍS Y EN LA CIUDAD DE LOJA

4.1.1 Sistemas para fotorradars en el país

La primera ciudad en implementar el sistema de fotorradars en Ecuador fue Quito, el 12 de enero del 2015 y, en agosto del mismo año se inició la instalación de los equipos en la ciudad de Loja [3], le siguieron Guayaquil y Cuenca que adoptaron también el sistema de multas de tránsito por medios electrónicos. De acuerdo a la legislación nacional en temas de tránsito, es necesario que los equipos que se usan para los sistemas de fotorradars estén debidamente homologados, ya que esto garantizara que se han realizados las respectivas revisiones y por ellos su óptimo funcionamiento. En la tabla 1 se muestran los equipos homologados para sistemas de multas por medio de fotorradars.

Tabla 1. Fotorradars homologados. [4]

FOTORRADARS						
NOMBRE DE LA EMPRESA O INSTITUCIÓN	MARCA	MODELO		No. DE CERTIFICADO	CLASIFICACIÓN	FECHA DE EMISIÓN DE HOMOLOGACIÓN
SERVICIO INTEGRADO DE SEGURIDAD ECU 911 (AZUAY)	CEIEC	Cámara:	TC-T207-C1-EGDE-ZDE	ANT-DRTTTSV-2013-FR-001	FIJO	04/06/2015
		Lector de placas:	TC-T820GE-EGDE-ZDE			
ALCOLISTI S.A.	SAFEPACE	CAM 400		ANT-DRTTTSV-2016-FR-002-R	FIJO	16/06/2016
		CAM 400 M		ANT-DRTTTSV-2016-FR-009	MÓVIL	02/05/2016
AGENCIA NACIONAL DE TRÁNSITO	ROBOT	MULTIRADAR C		ANT-DRTTTSV-2013-FR-003	ESTÁTICO	30/07/2014
SAFETY ENFORCEMENT SEGURIDAD VIAL S.A.	REDSPEED	SPEEDCURB		ANT-DRTTTSV-2016-FR-008	FIJO	20/10/2016
	ARH	CAM S1		ANT-DRTTTSV-2017-FR-008	MÓVIL	10/11/2017
	REDSPEED INTERNATIONAL LIMITED	REDSPEED 2.0		ANT-DRTTTSV-2017-FR-012	FIJO	13/11/2017
TELVENT TRÁFICO Y TRANSPORTE S.A.	EYP SCAP	RAI PLUS		ANT-DRTTTSV-2015-FR-006	MÓVIL	02/04/2015

PROTEKPROIN CIA. LTDA.	KUSTOM SIGNALS	LASERCAM 4	ANT-DRTTTSV-2015-FR-007	MÓVIL	03/10/2015
MANTENIMIENTO O TÉCNICO ESPECIALIZADO MTE CIA- LTDA.	TATTILE SRL	VEGA SPEED	ANT-DRTTTSV-2017-FR-010	FIJO	31/01/2017
TRAFFCONEC S.A.	GATSO	T-SERIES	ANT-DRTTTSV-2017-FR-011	FIJO - ESTÁTICO	09/05/2017
TRANSCORPECUADOR S.A.	IMAGSA TECHNOLOGIES S.A.	ATALAYA 3D	ANT-DRTTTSV-2017-FR-013	FIJO – MÓVIL	24/11/2017
IMPORVALIA S.A.	JENOPTIK-ROBOT	MULTIRADAR CD	ANT-DRTTTSV-2018-FR-0014	ESTÁTICO	01/03/2018
	JENOPTIK-ROBOT	TRAFFISTAR SR590	ANT-DRTTTSV-2018-FR-0015	FIJO	01/03/2018

4.1.2 Sistemas de fotomultas en la ciudad de Loja

Para la ciudad de Loja, el municipio contrato a la empresa *Safety enforcement seguridad vial*, Esta empresa, desde la segunda semana del mes de septiembre del 2015 empezó a generar los primeros resultados de fotomultas automáticamente [5]. Las cámaras usadas son de tecnología RedSpeed, las cuales son de tipo fijo y móvil; para un monitoreo constante durante las 24 horas del día pero con menor precisión y la otra con características móviles que permiten un monitoreo por cortos periodos a mayor precisión.

4.1.2.1 Cámaras fijas



Figura 1. Cámara Speedcurb. [6]

En Loja se encuentran instaladas cámaras marca RedSpeed modelo speedcurb [4], que además es un equipo para el control de velocidad, la cual usa sensores instalados en el asfalto para medir la velocidad de un vehículo en un punto específico.

La desventaja que presenta este equipo radica en que no puede medir velocidad a distancia, si no que realiza el cálculo por medio de los sensores ubicados en el asfalto. [7]

4.1.2.2 Cámaras móviles



Figura 2. Cámara por radar móvil ARH S1. [6]

El modelo de cámaras instaladas en la ciudad de Loja es ARH S1, este equipo cuenta con la principal ventaja que permite un escaneo de la vía de hasta 150 metros durante el día y en la noche de hasta 100 metros, además cuenta con un zoom de alto alcance, su desventaja es que solo puede operar durante un corto periodo de tiempo. [7]

4.1.2.3 Cartel informativo

A lo largo de las vías en las cuales se encuentran los sistemas de fotomultas, podemos ver algunos carteles electrónicos en los cuales se puede visualizar la velocidad a la que el vehículo se está desplazando en ese momento, este cartel utiliza un sensor de infrarrojo para esta tarea, la utilidad de este cartel es enteramente preventiva y la empresa encargada de la instalación de estos dispositivos es KRADAC, los carteles utilizan un sistema de energía basados en paneles solares para reducir el consumo energético



Figura 3. Indicador de velocidad. [8]

4.2 FUNCIONAMIENTO DE SISTEMA DE FOTOMULTAS EN LA CIUDAD DE LOJA

4.2.1 Funcionamiento básico del sistema de fotomultas en la ciudad Loja y notificación a infractores

4.2.1.2 Funcionamiento del sistema de fotomultas mediante cámaras fijas

El sistema proporcionado por la empresa SAFETY de cámaras fijas usa tecnología digital para realizar las fotografías, los equipos se instalan al costado de la carretera y utilizan sensores piezoeléctricos empotrados en el asfalto para la detección de velocidad de los vehículos.

El sistema toma tres fotografías del vehículo que ha excedido la velocidad, la primera es mediante zoom mostrando la matrícula del vehículo, la segunda intenta mostrar la marca, modelo del vehículo y fecha, la tercera muestra al vehículo en un instante de tiempo diferente a otra foto del mismo vehículo para hacer referencia. [9]

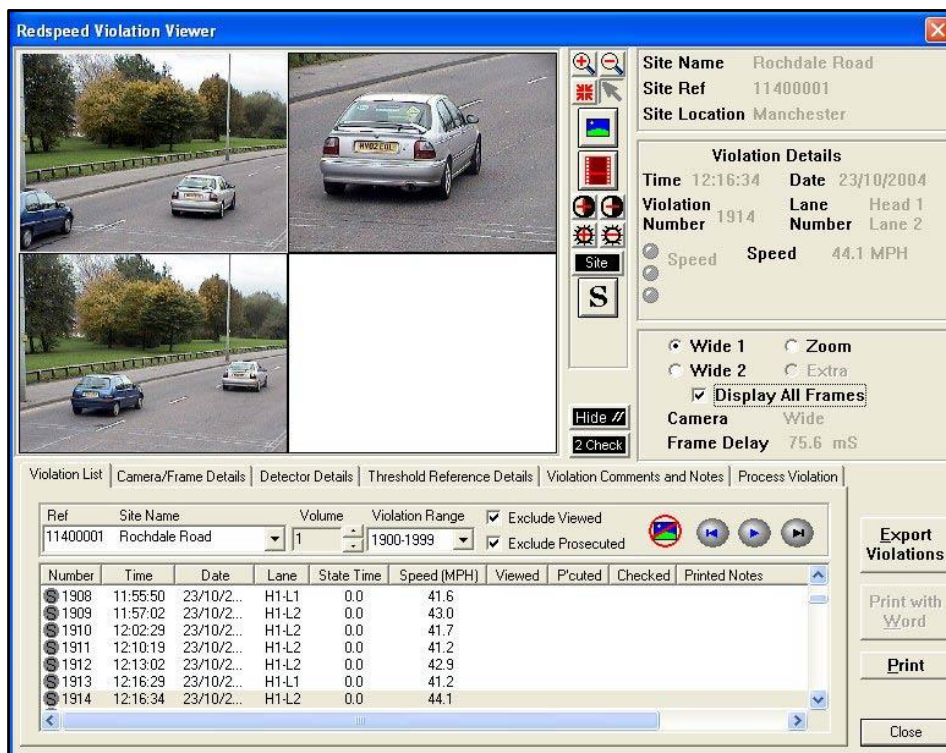


Figura 4. El sistema detecta tres fotografías del vehículo infractor. [9]

La velocidad se calcula utilizando la distancia entre sensores almacenada en la memoria del sistema, esta medida es comparada con una velocidad programada en el sistema. De existir el exceso de velocidad se procede a realizar las fotografías.

La altura de las cámaras, distancia del letrero informativo de velocidad, distancia de sensores piezoeléctricos, deben cumplir la normativa internacional del Manual on Uniform Traffic Control Devices (MUTCD) que establece en su apartado D4956 referente a especificaciones para revestimiento retrorreflectantes y materiales para el control de tránsito. En la siguiente tabla se muestran los datos que el sistema debe tener en consideración para la obtención de imágenes. En el ejemplo se muestran los valores obtenidos para una cámara con una distancia focal de 8 mm, ubicada a 12.2 metros de altura. [10]

Tabla 2 Ecuaciones y parámetros para la obtención de imágenes para Speedcurb. [10]

Dimensión	Símbolo	Cómo se calcula	valor
Altura de la cámara sobre la zona de detección	h	<i>es un parametro de entrada conocido</i>	12.2 metros
La mitad del ángulo de FOV vertical	α_V	<i>especificaciones de las lentes</i>	16.7 grados
La mitad del ángulo de FOV horizontal	α_H	<i>especificaciones de las lentes</i>	21.8 grados
Ángulo entre la estructura que sujeta la cámara y la línea de visión de la misma	θ	$\theta = (90 - 5) - \alpha_V$: donde el valor 5 es el numero de grados que el campo de visión esta por debajo del horizonte	68.3 grados
Distancia de la zona inferior de la imagen	d_1	$d_1 = h \tan(\theta - \alpha_V)$	15.4 metros
Anchura de la parte inferior de la imagen		$W_1 = 2 h \frac{\tan(\alpha_H)}{\cos(\theta - \alpha_V)}$	15.7 metros

	W_1		
Distancia a la parte superior de la imagen	d_2	$d_2 = h \tan (\theta + \alpha_V)$	139.4 metros
Anchura de la parte superior de la imagen	W_2	$W_2 = 2 h \frac{\tan (A_H)}{\cos (\theta + A_V)}$	112 metros

Los equipos también deben estar sujetos al reglamento de la ley de transporte terrestre, en donde se manifiestan límites de velocidad y tiempo en que se debe notificar al infractor correspondientes al artículo 191 y 238 respectivamente. [11]



Figura 5. Sensores Piezoeléctricos en asfalto. Fuente: El autor.

4.2.1.2 Funcionamiento de los carteles informativos

Los carteles informativos también forman parte de este sistema, como parte preventiva, este indica cuando se debe bajar la velocidad, los límites actuales en la ciudad de Loja impuesto en la ordenanza 0028A-2016 cumplen con lo establecido por la ANT en el artículo 190 y 191. Los límites de velocidad son los siguientes:

1). Velocidad para vehículos livianos, motocicletas y similares:

Tabla 3. Límites para vehículos livianos. [13]

Tipo de vía	Límite máximo	Rango moderado (art 142.g de la Ley)	Fuera del rango moderado (art 145.e de la ley)
Urbana	50 km/h	50 km/h-< 60 km/h	60 km/h
perimetral	90 km/h	90 km/h-< 120 km/h	120 km/h
Rectas en carreteras	100 km/h	100km/h-<135km/h	135 km/h
Curvas en carreteras	60 km/h	60 km/h-< 75km/h	75 km/h

2). Velocidad para vehículos de transporte público de pasajeros:

Tabla 4. Límites de velocidad para transporte público. [13]

Tipo de vía	Límite máximo	Rango moderado (art 142.g de la Ley)	Fuera del rango moderado (art 145.e de la ley)
Urbana	40 km/h	40 km/h-< 50 km/h	50 km/h
perimetral	70 km/h	70 km/h-< 100 km/h	100 km/h
Rectas en carreteras	90 km/h	90 km/h-<115km/h	115 km/h
Curvas en carreteras	50 km/h	50 km/h-< 65km/h	65 km/h

3). Velocidad para vehículos de transporte de carga:

Tabla 5. Límites de velocidad para transporte de carga. [13]

Tipo de vía	Límite máximo	Rango moderado (art 142.g de la Ley)	Fuera del rango moderado (art 145.e de la ley)
Urbana	40 km/h	40 km/h-< 50 km/h	50 km/h
perimetral	70 km/h	70 km/h-< 95 km/h	95 km/h
Rectas en carreteras	70 km/h	70km/h-<100km/h	100 km/h
Curvas en carreteras	40 km/h	40 km/h-< 60km/h	60 km/h

Los carteles utilizan el efecto Doppler para medir la velocidad de un vehículo, además cuentan con un panel fotovoltaico y una batería interna que sirve como suministro durante la noche. Cuando la energía cae por debajo de 10.5 V un controlador envía la señal para que el sistema se apague y envía otra señal para restablecerlo cuando este voltaje sube por encima de 12.5 V. el ancho de haz del radar es de 40 a 30 grados, aunque se puede ajustar de 12 a 10 para casos especiales y pueden mediar velocidades desde 0 a 200 km/h. [12]

En cartel informativo debe ubicarse como se muestra en la figura 6 para tener su máximo rendimiento.

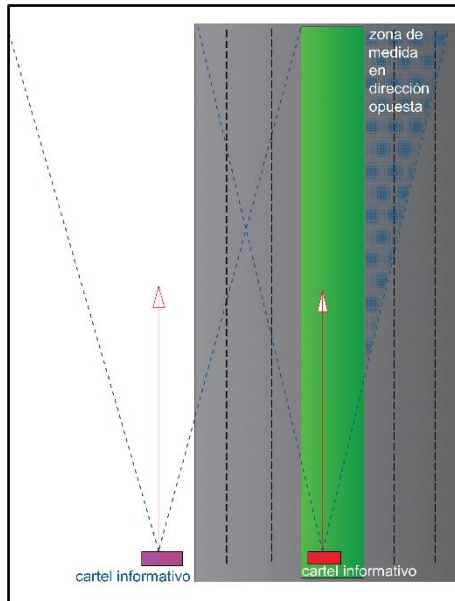


Figura 6. Ubicación correcta del cartel informativo. [12]

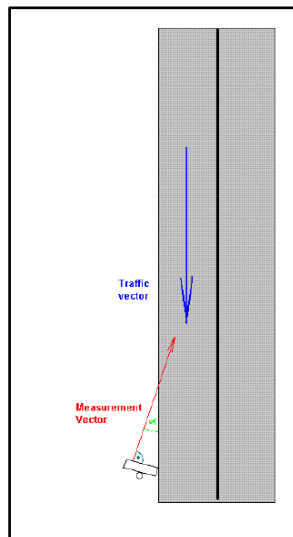


Figura 7. Ubicación con ángulo del cartel informativo. [12]

Cuando el cartel se ubica con ángulo respecto a la carretera, como se muestra en la figura 7, el cartel no detecta la velocidad al 100% si no que el controlador multiplica la velocidad obtenida por el ángulo Alpha.

4.2.1.3 Notificación a los posibles infractores

Luego que los sensores instalados en el piso detectan una infracción y las cámaras fotografían al vehículo. El infractor recibirá una notificación de la sanción a su correo electrónico, ya que todos los usuarios deben llenar como requisito su correo electrónico a la hora de matricular su vehículo. Desde el momento en que la notificación es recibida,

se tiene tres días (72 horas) para apelar dicha sanción. Si, dentro de 72 horas no se ha revisado el correo electrónico y se lo ha hecho después, la Agencia Nacional de Tránsito (ANT) entiende que la multa fue aceptada. Si la multa es levantada mediante radar móvil, la notificación es recibida personalmente por agentes de tránsito [14]

4.3 FUNCIONAMIENTO DEL SISTEMA DE POSICIONAMIENTO GLOBAL (GPS).

La propuesta de este proyecto plantea el uso de la tecnología para la obtención de velocidad, posición y hora, la cual se almacenará y enviará mediante GSM/GPRS para un uso a priori.

La tecnología del GPS tiene como principal característica la triangulación, esta mide el tiempo de propagación de las señales de radio. La distancia entre el GPS y el satélite se obtiene multiplicando el tiempo de la señal expuesta desde el satélite multiplicado por su velocidad de propagación, esto se detalla matemáticamente más adelante [15, p. 34]

4.3.1 Composición del sistema de GPS.

Se encuentra constituido de tres partes:

- ✓ Satélites
- ✓ Receptores
- ✓ Control terrestre

El sistema lo componen 24 satélites que se encuentran distribuidos en 6 orbitas polares a 2170 kilómetros de la tierra, los satélites orbitan la tierra de tal forma que siempre se encuentran visibles por lo menos 4 satélites para los receptores de GPS, por norma general y para mayor exactitud siempre se encuentran visibles 8 satélites para cualquier receptor

El control terrestre se realiza mediante 5 estaciones distribuidas al largo del planeta, estas monitorean operación y posición en el espacio, además realizan actualizaciones en el software. Las estaciones que existen actualmente son la de HAWAI, Isla ASCENSION, DIEGO GARCIA, KWAJALEIN y COLORADO SPRINGS [15, p. 31]

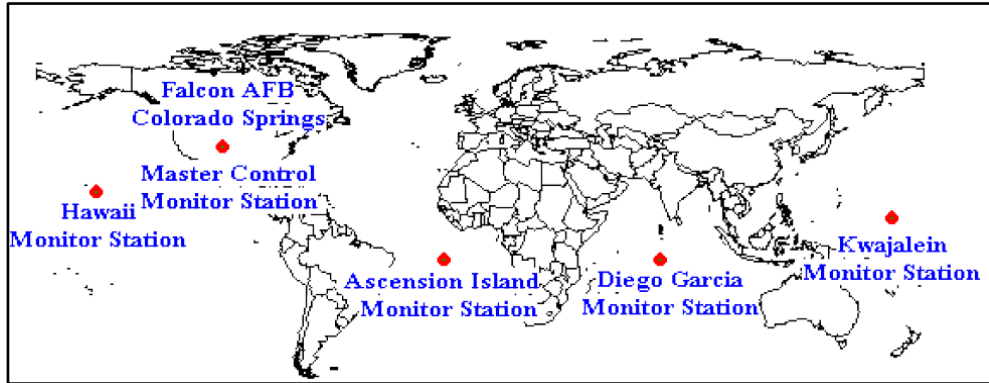


Figura 8. Estaciones terrenas para control de GPS en todo el mundo. [16]

4.3.2 Principio matemático del GPS

La señal emitida por un satélite llega al receptor terrestre a la velocidad de la luz, ese valor multiplicado por el tiempo que esta señal llega a la tierra nos da como resultado la distancia a la que se encuentra el satélite.

Con tres satélites se podría determinar la posición en la que nos encontramos. Es necesario para la obtención de ecuaciones el uso de cuatro satélites para medir sus distancias. En la figura 9 podemos ver cómo funciona la trilateración.

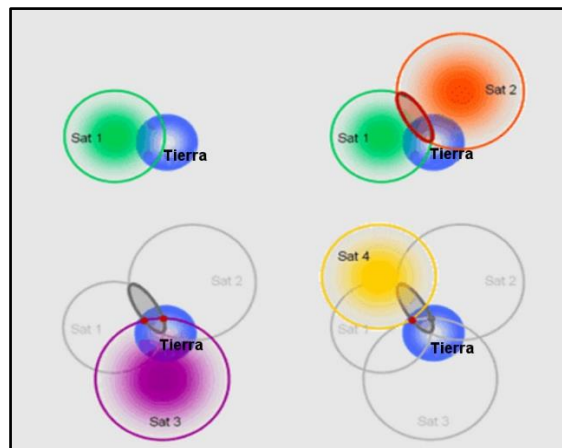


Figura 9. Forma grafica de la trilateracion para obtener un punto determinado en la tierra. [17]

Recibiendo la señal de un solo satélite, podemos obtener la posición en la que nos encontramos en cualquier punto de la superficie de la tierra de radio R1

Con la señal de un segundo satélite podemos reducir la región en la que nos encontramos a la intersección de dos esferas de radio R1 y R2. Mediante la señal de un tercer satélite podemos reducir la zona a 2 puntos y mediante un cuarto satélite podemos determinar la solución en un solo punto como se muestra en la figura 9.

Matemáticamente:

El tiempo en que el satélite envía la señal es t_0 y sabemos el tiempo en el que hemos recibido la señal t_1 ; para saber el tiempo en que la señal se ha demorado en llegar a nosotros obtenemos el diferencial de estos tiempos

$$\Delta_t = t_0 + t_1 \text{ (Ec. 1) [16]}$$

Para calcular la distancia a la que se encuentra el satélite; multiplicamos el tiempo que se demora en llegar la señal por la velocidad a la que viaja: tendríamos, $r = |r| = c * \Delta_t$, donde c es la velocidad de la luz y r es la distancia de receptor al satélite. Se define también que:

$$\rho = c * \Delta_t \text{ (Ec. 2) [16]}$$

También existe el error en las mediciones mediante GPS. Debido al sincronismo que normalmente sufre algún desfase entre el receptor y el satélite. El satélite contiene un reloj de cesio, pero el reloj del GPS es de cuarzo y está sujeto a errores; esto lo determinamos de la siguiente manera:

$$t_{receptor} = t_{GPS} + t_u \text{ (Ec. 3)}$$

donde t_u es el pequeño error, dado que el valor de ρ no es una medida de distancia exacta, se le conoce como *pseudodistancia*. [16]

Luego, suponiendo que u es nuestra posición y s la posición del satélite, mediante esto podemos obtener la distancia del receptor al satélite asumiendo que:

$$r = s - u \text{ (Ec. 4)}$$

Después de esto podemos agrupar las ecuaciones Ec. 2 y Ec. 4 descritas anteriormente, tenemos que:

$$\rho - c * t_u = |s - u| \text{ (Ec. 5)}$$

Podemos notar que la ecuación descrita anteriormente determina la forma de una esfera y dado q se trata de 4 satélites, tenemos [16] :

$$\rho_1 - ct_u = |s - u| \text{ (Ec. 6)}$$

$$\rho_2 - ct_u = |s - u| \text{ (Ec. 7)}$$

$$\rho_3 - ct_u = |s - u| \text{ (Ec. 8)}$$

$$\rho_4 - ct_u = |s - u| \text{ (Ec. 9)}$$

Al resolver las siguientes ecuaciones:

$$\rho_1 = \sqrt{(x_1 - x_u)^2 + (y_1 - y_u)^2 + (z_1 - z_u)^2} + ct_u \text{ (Ec. 10)}$$

$$\rho_1 = \sqrt{(x_2 - x_u)^2 + (y_2 - y_u)^2 + (z_2 - z_u)^2} + ct_u \text{ (Ec. 11)}$$

$$\rho_1 = \sqrt{(x_3 - x_u)^2 + (y_3 - y_u)^2 + (z_3 - z_u)^2} + ct_u \text{ (Ec. 12)}$$

$$\rho_1 = \sqrt{(x_4 - x_u)^2 + (y_4 - y_u)^2 + (z_4 - z_u)^2} + ct_u \text{ (Ec. 13)}$$

4.3.3 Formatos GPS

Existen varios formatos para la tecnología de GPS. Los formatos permiten establecer el uso de datos y la interfaz de diferentes receptores. Debido a la cantidad de fabricantes se han estandarizado algunos. Entre los principales tenemos: NMEA, RTCM, RINEX, NGS-SP3

4.3.3.2 Estándar NMEA-2000

La norma para la Interfaz de Dispositivos Electrónicos Marinos o por sus siglas en ingles Standard For Interfacing Marine Electronic Devices (NMEA), define los parámetros electrónicos de señalización, protocolo de transmisión de datos y formato de sentencias, en su primera versión opera a una velocidad de 4800 Baudios, en su nueva versión mejorada conocida como NMEA 2000 maneja velocidad de 9600 Baudios [18]

Para este proyecto se está usando este tipo de formato ya que es manejado por GPS's de bajo costo y además la información que presenta sus sentencias (tramas GPS) son las requeridas para la obtención de datos de velocidad en Km/h, hora, fecha y posición.

Entre las sentencias que proporciona este tipo de formato tenemos las siguientes:

- ✓ GPRMC (Recommended Minimum Specific). – aquí se encuentra la información de posición, latitud, longitud, fecha, hora y variación magnética.
- ✓ GPGGA (Global Positioning System Fixed Data). – aquí se encuentra la información sobre cantidad de satélites en uso.

- ✓ GPGSA (GNSS (Global Satellite navigation System) DOP (Dilution of Precision) and Active Satellites). – información referida al indicador de satélites en uso.
- ✓ GPGSV (GNSS Satellites in View) aquí se encuentra la información sobre los satélites como elevación, azimut, SNR. Etc.

4.4 GSM Y GPRS

4.4.1 Red para el Sistema global para las comunicaciones móviles o Global System for Mobile Communications (GSM)

La red GSM es una red de segunda generación, esta tecnología utiliza acceso múltiple por división de tiempo o Time Division Multiple Access por sus siglas en inglés (TDMA), se inició en Europa en el año 1992. Usa transmisión digital y no limita únicamente a transmisión de voz, ya que permite transmisión de texto y datos con una velocidad de hasta 14.4 Kbps [19]

4.4.2 Servicio general por paquetes vía radio o general Packet Radio Service (GPRS)

GPRS es una técnica de conmutación por paquete, integrable con la estructura actual de GSM ya que es una extensión de este sistema, esta tecnología permite una velocidad de datos de 144 Kbps, no necesita un circuito dedicado para cada usuario conectado. [20]

Mientras GSM basaba su cobertura a las comunicaciones móviles mediante conexiones satelitales, GPRS funcionaba de igual modo, con la diferencia que el sistema usaba conexiones de radio. [21]

4.4.3 Ventajas de GPRS

Entre las principales ventajas tenemos:

- ✓ Permite enviar y recibir información IP
- ✓ Utiliza conmutación por paquetes
- ✓ Utiliza el canal de comunicación por demanda, lo que indica que es usado solo cuando se lo necesita, por lo cual el canal queda libre para otros usuarios
- ✓ Dado que es una extensión de GSM es compatible con LAN, WAN e INTERNET

- ✓ Tasa de transmisión de 56 a 144 Kbps
- ✓ Permite realizar llamadas GSM y al mismo tiempo enviar información GPRS. [19]

4.5 BASE DE DATOS.

4.5.1 Concepto de base de datos

Una base de datos es un sistema formado por un conjunto de información almacenada en discos que permiten el acceso directo a ella y un grupo de programas que manipulen ese cúmulo de datos [22] .

4.5.2 Tipo de base de datos.

Existen varios tipos de base de datos que se describen a continuación: según la forma en la que van variando y según el contenido que posee.

- **Según la forma en la que van variando.** - dentro de este tipo podemos hablar de dos tipos.
 - 1) Estática. - Son bastante usadas en el mundo empresarial y registro histórico. En este tipo de base de datos la información no puede ser modificados por o que se puede decir que esta información es de solo lectura.
 - 2) Dinámica. - La información que se almacena bajo base de datos dinámica posee la principal característica que la información si puede ser modificada y borrada de manera ordenada.
- **Según el contenido que maneja.** – dentro de este tipo de base de datos se manejan tres sub tipos.
 - 1) Bibliográficas. – La información que se maneja en este tipo de base de datos contiene datos acerca de material bibliográfico.
 - 2) Directorios. – Aquí se guarda información personalizada de correos electrónicos, o números telefónicos.
 - 3) Especializada. – aquí se maneja información técnica basada en las necesidades puntuales de consumidores de dicha información.

4.5.3 Base de datos en forma de texto

Una base de datos manejada en forma de texto puede ahorra gran cantidad de trabajo, ya es fácilmente editable.

Una aplicación web basada en texto dinámico, que puede ser reescrita y borrable posee como principal característica que no representa mayor dificultad para ser subida al sitio en el que se quiere visualizar. Si el texto se maneja como archivo CSV, es de fácil acceso y no necesita de aplicaciones adicionales para manipular dicha información, esta podría manejarse mediante un bloc de notas, por ejemplo. [23]

5. MATERIALES Y MÉTODOS

El presente trabajo tiene como objeto el desarrollo de un circuito electrónico adaptable a un vehículo de transporte público o privado que mediante un sistema de GPS podrá obtener la velocidad, ubicación, hora y fecha del mismo, creando un registro cada 5 segundos y enviando dicha información mediante un módulo GPRS/GSM a una PC que en este caso funcionará como base de datos.

5.1 Metodología.

Este proyecto se basa en el método científico deductivo, el cual, conceptualmente expresa que el pensamiento puede ir de leyes y principios a hechos concretos. Para este proyecto se tomaron los principios generales de la tecnología de varios componentes como: GPS, GSM, procesamiento de datos y la teoría manejada por los diferentes elementos individuales; como resistencias, capacitores, semiconductores entre otros. Para luego de esto llegar a una idea concreta establecida en los diferentes objetivos

5.2 DISEÑO DEL SISTEMA PARA LA OBTENCIÓN DE POSICIÓN, HORA Y VELOCIDAD.

5.2.1 Desarrollo cronológico del sistema para la obtención de posición, hora, velocidad y envío de datos.

El sistema electrónico para la Medición y Registro de Velocidad de un Vehículo Mediante GPS se explica a través del diagrama de bloques que se muestra en la figura 10. El mismo está conformado por un sistema de control basado en un microcontrolador, sistema de posicionamiento que consta de un módulo GPS, sistema de comunicación (que usa módulos GSM/GPRS para transmisión y recepción de la información) y base de datos.

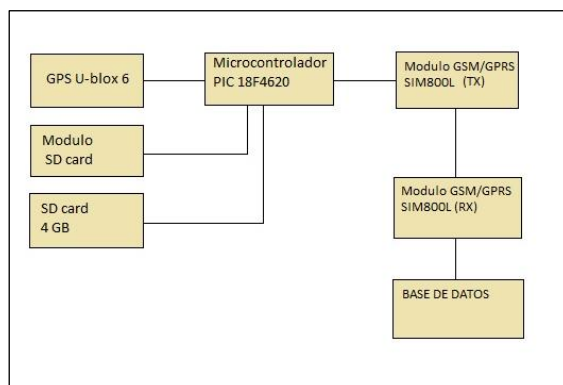


Figura 10. Diagrama de bloques del Sistema Electrónico para la Medición y Registro de Velocidad. Fuente: El autor.

En la figura 10 se puede evidenciar que el diseño del Sistema Electrónico para la Medición y Registro de Velocidad de un Vehículo Mediante GPS se enmarca en el requerimiento de un conjunto de dispositivos que permitan de manera adecuada recopilar, procesar, transmitir y almacenar la información recopilada de los diferentes módulos y elementos necesarios para el escenario planteado en el objetivo general del presente trabajo. Como primera parte, se seleccionará el elemento de procesamiento, que será un microcontrolador PIC, el cual se encarga de procesar los datos obtenidos por un módulo GPS debidamente seleccionado, además deberá procesar estos datos para mostrarlos en un Display, luego de esto la información obtenida y actualizada cada 5 segundos es guardada en una memoria SD, para posteriormente enviar esta información a través de tecnología GSM/GPRS de la red celular de alguna operadora hacia una base de datos previamente establecida. El sistema diseñado estará en la capacidad de medir velocidad, obtener posición, hora, procesar y almacenar la información obtenida para finalmente enviarla hasta una computadora.

El orden cronológico mediante el cual se desarrolló este dispositivo fue el siguiente:

- ✓ Selección de los diferentes módulos que conforman el Sistema Electrónico para la Medición y Registro de Velocidad de un Vehículo Mediante GPS.
- ✓ Montaje de los elementos para el procesamiento (microcontrolador y pantalla LCD) y montaje del módulo GPS para obtención de datos
- ✓ Almacenamiento de los datos obtenidos en una memoria SD de 4G
- ✓ Envío de la información obtenida a través de un módulo GSM/GPRS
- ✓ Almacenamiento de la información en una base de datos
- ✓ Visualización de la información a través del receptor en la PC

La información almacenada en la base de datos a más de ser información útil para apelar una posible multa por exceso de velocidad además puede ser usada a futuro para una integración con varios dispositivos y así tener un registro del recorrido que han realizado varios vehículos en la ciudad de Loja.

5.3 SELECCIÓN DE LOS COMPONENTES

5.3.1 Selección del Microcontrolador

En la selección del microcontrolador se tomó en cuenta algunos factores; como: precio, disponibilidad en el mercado, cantidad de memoria, puertos seriales, procesamiento entre

otros. Dentro de los microcontroladores que se estudiaron para la ejecución de este proyecto se consideró:

5.3.1.1 Arduino uno

El Arduino Uno es un tablero de microcontroladores basados en ATmega328 que es un microcontrolador que consta de 14 pines digitales de entrada y de salida, consta además de 6 entradas analógicas, además tiene entradas para varios periféricos, tiene un oscilador de 16 MHz, cuenta también con un botón de reinicio. [23]

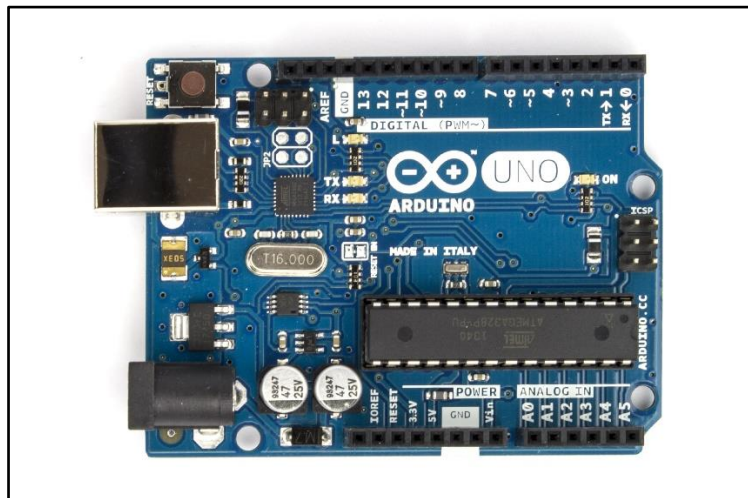


Figura 11. Módulo arduino UNO. [23]

Características generales:

- | | |
|---------------------------------|------------------------|
| ✓ Microcontrolador | ATmega328 |
| ✓ Voltaje de funcionamiento | 5V |
| ✓ Voltaje de entrada | 7-12V |
| ✓ Pines digitales I/O | 14 (6 para salida PWM) |
| ✓ Pines de entrada analógica | 6 |
| ✓ Corriente DC por pin I/O | 40 mA |
| ✓ Corriente DC para pin de 3.3V | 50 mA |
| ✓ Memoria flash | 32 Kb (ATmega328) |
| ✓ SRAM | 2 Kb (ATmega328) |
| ✓ EEPROM | 1Kb (ATmega328) |
| ✓ Cristal | 16 Mhz |

5.3.1.2 Desventajas del Arduino Uno:

- ✓ La programación se la hace en *ensamblar* y el precio a pagar por las librerías es algo costoso.
- ✓ El módulo Arduino Uno viene ya ensamblado lo que resta flexibilidad a los proyectos. Los PCB's deben armarse tomando en cuenta el tamaño del módulo mencionado.
- ✓ Los módulos adicionales suelen ser costos

5.3.1.3 Ventajas de Arduino Uno:

- ✓ Simplifica la ejecución de proyectos
- ✓ Se puede ampliar un proyecto con la adquisición de más módulos
- ✓ Multi-plataforma, debido a que Arduino es compatibles con algunos sistemas operativos como Windows y Linux.

5.3.1.4 Microchip familia 16FXX

Los microcontroladores de la familia PIC (Controlador de Interfaz Periférico) 16F y 18F son de gran robustez, ya que cuentan con una gran capacidad de memoria y capacidad de procesamiento, son de bajo y muy útiles para el desarrollo de proyectos, además puede ser programado en lenguajes como. BASIC, Lenguaje C, Ensamblador

5.3.1.5 PIC 16F628A

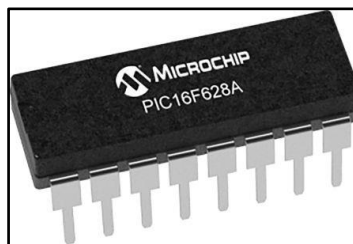


Figura 12. Microcontrolador PIC16F628A. [24]

El PIC 16F628A pertenece a la familia de microcontroladores de bajo costo de 8 bits. Son los más populares de la familia 16F, en la actualidad el modelo 16F628A soporta más de 100.000 ciclos de escritura en su memoria flash y 1'000.000 de ciclos en la memoria EEPROM. La tecnología de PIC está basada en CMOS, por lo que son propensos a la

corriente estática, esto significa que deben tratarse con cuidado para evitar que el dispositivo se averíe

5.3.1.6 Características del PIC 16F628A

- | | |
|----------------------------------|---------------------------------|
| ✓ Máxima frecuencia de operación | 20 Hz (reloj) |
| ✓ Memoria flash | 2048 palabras |
| ✓ Memoria RAM | 224 bytes |
| ✓ EEPROM | 128 bytes (8 bits por registro) |
| ✓ Voltage de operación | 3 a 5.5 V |
| ✓ Comunicación digital | 1 Puerto UART 1 A/E/UART |
| ✓ Cantidad de terminales | 18 incluyendo alimentación |

5.3.1.7 Desventajas del PIC 16F628A

- ✓ Acceso a la pila (ya que es de tamaño fijo)
- ✓ Las instrucciones para este dispositivo son reducidas (35 instrucciones)
- ✓ Es sensible a la estática, el fabricante recomienda el uso de pulseras magnéticas o guantes para su manipulación

5.3.1.8 Ventajas del PIC 16F628A

- ✓ Bajo costo
- ✓ Gran cantidad de información por tratarse de los más populares
- ✓ Posee una arquitectura de tipo hardware
- ✓ Maneja un juego de instrucciones fijas, simple y reducido
- ✓ El software de programación para el 16F628A es de fácil acceso

5.3.1.9 PIC 18F4620



Figura 13. Microcontrolador PIC16F628A. Fuente: El autor

El PIC 18F4620 es un microcontrolador fabricado por Microchip de gama media, que dentro de sus características principales presenta un módulo controlador de LCD integrado, capaz de manejar 48 segmentos y 4 comunes para pantalla LCD (132 pixeles). A diferencia de otros modelos de PIC 18F4620 cuenta con una arquitectura en la que los puertos EUSART y el puerto SPI están separados independientemente [24]

5.3.1.10 características

- ✓ Hasta 10 MIPS performance a 3V
- ✓ Arquitectura compacta RISC del compilador C
- ✓ Multiplicación del Hardware de un solo ciclo 8x8
- ✓ Su oscilador interno soporta desde 31 KHz a 8 MHz
- ✓ Arquitectura de RISC del compilador C
- ✓ Rango de voltaje de operación de 2.0 V a 5.5 V
- ✓ Modulo controlador de LCD
- ✓ ADC de 10 bits
- ✓ Modulo EUSART que incluye soporte de bus LIN

5.3.1.11 Características adicionales

- | | |
|------------------------------------|------------------------------|
| ✓ Tamaño de la memoria programable | 64Kb |
| ✓ Bytes SRAM | 3.968 |
| ✓ Datos EEPROM | 1024 bytes |
| ✓ Periféricos de comunicación | 1-UART, 1-SPI, 1-I2C1-MSS... |
| ✓ Numero de puertos | 40 |

5.3.1.12 Ventas del PIC 18F4620

- ✓ Posee un módulo para LCD lo que lo hace muy versátil al momento de desarrollo de proyectos
- ✓ Su configuración interna no comparte pines dentro de los puertos EUSART Y SPI, por lo q los puertos pueden ser usados individualmente para tecnologías q operan mediante puertos seriales
- ✓ Cuenta con una gran cantidad de memoria para el desarrollo de proyectos 64 Kb para programar en el dispositivo

5.3.1.13 Desventajas del PIC 18F4620

- ✓ Sensible a la estática
- ✓ Puertos seriales limitados para el uso de proyectos

5.3.1.14 Microcontrolador elegido

En la tabla 7 se detallan características principales de los diferentes microcontroladores que se tomaron en cuenta para este proyecto.

Tabla 6. Comparación de los diferentes microcontroladores estudiados para este proyecto. Fuente: El autor.

prestaciones	Arduino UNO	PIC 16F628A	PIC 18F4620
microcontrolador	ATmega328V	16F628A	18F4620
Voltaje de entrada recomendado	5V	3-5.5V	2-5.5V
Pines digitales	14 pines de los cuales 6 para PWM	8 pines	16 pines
Pines analógicos	6 pines	8 pines	16 pines
SPI	si	si	si
USART	si	si	si
documentación	si	si	si
Entorno de desarrollo	ARDUINO 1.8.9	MikroC	MikroC
precio	25 \$	4 \$	7\$
Arquitectura	No permite modificaciones	Admite hardware adicional	Admite hardware adicional
Memoria de almacenamiento	Páginas de 1 Kbyte	Páginas de 2 Kbyte	Páginas de 4 Kbyte

En la tabla 6 se puede evidenciar algunas características importantes que se tomaron en cuenta para el desarrollo de este proyecto, entre las cuales destacan la arquitectura, voltaje de operación, memoria y precio entre otras características relevantes.

Adicional a esta tabla se puede mencionar la disponibilidad en el mercado de cada uno de los microcontroladores mencionados. Cabe destacar el alto precio del módulo Arduino UNO, además que su arquitectura no permite modificaciones ya que viene desarrollado y su baja memoria de almacenamiento no permite un óptimo funcionamiento para este proyecto, por otra parte tenemos al PIC de la familia 16F, que cuenta con 2 Kb de memoria y un bajo precio, en este caso se estudió la posibilidad de usar este microcontrolador, pero presenta la desventaja que maneja 20 pines y tomando en cuenta que la cantidad de periféricos que maneja este proyecto (módulos, GPS, GSM, SD Card) requieren de un microcontrolador que maneje mayor cantidad de dispositivos externos (por lo tanto mayor cantidad de pines), memoria y que maneje mayor cantidad de puertos USART, por lo cual el PIC 18F4620 cumple los requerimientos necesarios para manejar la cantidad de módulos mencionados para este proyecto, además la memoria es suficiente ya que permite manejar las instrucciones que debe ejecutar (4 Kb), es de bajo precio y cuenta con documentación suficiente para trabajar sobre este microcontrolador.

Por lo tanto, el microcontrolador elegido para este proyecto fue el PIC 18F4620, principalmente por sus puertos que no encuentran multiplexados en su arquitectura interna, permitiendo la conexión con varios periféricos como: modulo GPS, modulo GSM/GPRS y el módulo para Micro SD, su capacidad de memoria y puertos USART.



Figura 14. Microcontrolador PIC18FF4620. [25]

5.3.2 Selección del dispositivo GPS

Para la elaboración de este proyecto se analizaron algunos módulos GPS. Considerando su costo, disponibilidad en el mercado local, precisión, compatibilidad con el microcontrolador, arranque del dispositivo, trama manejada y el formato usado por el módulo.

5.3.2.1 Receptor GPS P28146 DE Parallax

Los datos que proporciona este módulo, son los siguientes: latitud, longitud, fecha (incluida hora), además permite obtener el sentido en que se está desplazando el receptor

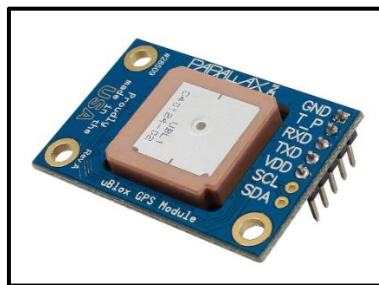


Figura 15. Modulo GPS P28146. [26]

Este módulo cuenta con orificios para su fácil fijación en tableros correspondiente a proyectos sin necesidad de soldadura. Es ideal para usarlo con microcontroladores que funcionen a 3.3 a 5 V

5.3.2.2 Características técnicas [26]

- ✓ Protocolo estándar NMEA 0183 v2.3
- ✓ 10Hz de velocidad de actualización de navegación
- ✓ Error en la precisión horizontal menor a 2.5 metros
- ✓ Señal de pulso de tiempo configurable de 0.25 a 1 KHz
- ✓ Comunicación serial UART
- ✓ Altitud máxima de 50 Km (estratosfera superior)
- ✓ Batería asistida por supercondensador
- ✓ Voltaje de operación 3.3 a 5 V, requiere 55mA

5.3.2.3 Desventajas del GPS P28146

El módulo P28146 también presenta algunas desventajas que se detallan a continuación:

- ✓ Como principal desventaja tenemos que su antena se encuentra integrada al circuito del módulo P28146, lo que no permite manipular el circuito a su totalidad
- ✓ Otra desventaja importante es la poca información que existe sobre este dispositivo, aunque presenta características importantes, la poca información no permite una implementación más detallada
- ✓ Su precio es algo elevado, aproximadamente 50\$ en el mercado local

5.3.2.4 Ventajas del GPS P28146

Este módulo cuenta con algunas ventajas sobre otros dispositivos mencionadas a continuación:

- ✓ Posee un margen de error bastante aceptable
- ✓ Cuenta con un inicio asistido por un supercondensador
- ✓ Opera con niveles de voltaje compatibles con la mayoría de microcontroladores

5.2.2.5 Modulo GPS u-blox 6

La serie de módulos NEO-6 pertenece a los GPS autónomos que usan el rendimiento del *u-blox 6*. El motor de posicionamiento de 50 canales cuenta con un tiempo de corrección de menos un segundo. Su arquitectura permite un consumo bastante bajo de energía [27]



Figura 16. Modulo GPS u-blox 6. [27]

5.2.2.6 Características del módulo u-blox 6 GPS [27]

- | | |
|---------------------------------------|------------------------|
| ✓ Voltaje de alimentación | desde -0.5 hasta 2.0 V |
| ✓ Voltaje de batería de respaldo | desde -0.5 hasta 3.6 V |
| ✓ Corriente de DC en cualquier puerto | 10 mA |
| ✓ Temperatura de almacenamiento | -40 hasta 85°C |
| ✓ Corriente de suministro máxima | 67 mA |
| ✓ Protocolo | NMEA, UBX, RTCM |

✓ Interfaz

UART, SPI

5.3.2.7 Ventajas del módulo GPS u-blox 6

- ✓ Conexión en menor tiempo que otros dispositivos
- ✓ Bajo costo
- ✓ Menor consumo de energía
- ✓ Puede operar bajo su plataforma online
- ✓ Bastante información sobre el dispositivo
- ✓ Cuenta con una batería para almacenar la información, lo permite su rápida conexión
- ✓ Su antena no viene adherida al módulo, lo que permite mayor libertad al momento de desarrollar proyectos

5.3.2.8 Desventajas del módulo GPS u-blox 6

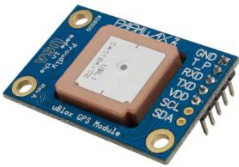

Existen pocas desventajas para este dispositivo, ya que es ideal para proyectos de esta naturaleza por contar con grandes prestaciones descritas anteriormente. Entre las pocas desventajas tenemos:

- ✓ Necesita un espacio abierto para realizar la conexión con los satélites disponibles
- ✓ Es un dispositivo sensible a la electrostática, por lo que debe manipularse con precaución, ya que cuenta con una antena de parche

5.3.2.9 modulo GPS elegido.

En la tabla 7 se puede analizar las características principales que se debieron tomar en cuenta para la elección del módulo GPS, entre los principales factores a tomar en cuenta tenemos: voltaje de operación, arquitectura, procesamiento, tiempo de respuesta y su costo dentro del mercado local.

Tabla 7. Comparación de las propiedades entre GPS P28146 Y GPS u-blox 6. Fuente: El autor.

GPS'S	GPS P28146	GPS u-blox 6
prestaciones		
Voltaje de operación	3.3-5.5 V	-0.5-2.0 V
protocolo	NMEA 0183 v2.3	NMEA, UBX, RTCM
Error de medición	2.5 m	1.5 m
hardware	Poco adaptable	adaptable
precio	50\$	20\$
documentación	poca	si
Tiempo de respuesta	3min	4min

En la tabla 7 podemos notar varias características importantes de los módulos GPS estudiados. Para el desarrollo de este proyecto fue necesario tomar en cuenta tres factores necesarios, documentación, hardware y consumo, como se puede observar en la tabla 6, existe poca documentación sobre el GPS P28146 y opera con voltajes que van desde 3.3-5.5 además su hardware no es adaptable ya que su antena viene integrada en el circuito, por su parte el GPS U-blox 6, opera con voltajes que van desde -0.5 V hasta 2 V, lo que permite trabajar con salidas del microcontrolador programables, su antena no viene integrada lo que permite mayor operabilidad, existe gran cantidad de documentación sobre este módulo y cabe destacar su bajo precio. Por lo tanto, el módulo elegido fue el GPS U-blox 6

5.3.4 SELECCIÓN DEL MÓDULO GPRS

La función que cumple el módulo GPRS es la de enviar la información obtenida por el módulo GPS hasta una computadora. Aprovechando la función que tiene el módulo

GPRS de envío mediante paquetes o SMS. La información se almacenará en una computadora donde se dispondrá la información para poder ser accedida por el usuario

Se analizaron algunas características de dos módulos GPRS para obtener el más optimo y que se adapte a las necesidades de este proyecto.

5.3.4.1 Modulo GSM/GPRS SIM 900

Se trata de un módulo cuatribanda de la marca SIMCOM que cuenta con aplicaciones como voz y datos. Puede comunicarse a través de puerto serial para aplicaciones por medio de microcontroladores. El sistema cuenta con un stack TCP/IP por el cual puede aplicarse en áreas que requieran conexión a internet



Figura 17. Modulo GPRS SIM 900. [28]

5.3.4.2 Características del módulo GPS/GPRS SIM900 [28]

- ✓ Maneja cuatribanda GSM de 850/900/1800/1900 Hz
- ✓ Potencia de transmisión 2 W 850/900 Mhz
- ✓ Kit de herramientas de la aplicación SIM
- ✓ Rango de tensión de alimetación: 9 a 20 V
- ✓ Consumo de energía bastante bajo: 1mA
- ✓ Copia de seguridad RTC
- ✓ Clase 10 de GPRS: 85.6 Kbps (enlace descendente)

5.3.4.3 Ventajas del GSM/GPRS SIM 900

- ✓ Bajo consumo de corriente, lo que permite una integración a proyectos que manejen poca corriente
- ✓ Maneja cuatribanda, esto hace que pueda trabajar en varios lugares que manejen diferentes bandas

5.2.3.4 Desventajas del GSM/GPRS SIM 900

- ✓ Alto costo del dispositivo (53 \$ aproximadamente en el mercado local)
- ✓ Poca disponibilidad en el mercado local

5.3.4.5 Modulo GSM/GPRS SIM 800L

Es un módulo para comunicación mediante la tecnología GPRS. Permite el envío de texto, datos y voz, es de tamaño reducido y maneja cuatribanda. A continuación, se detallan varias características de este módulo. [29]



Figura 18. Modulo GSM/GPRS SIM800L. [29]

5.3.4.6 Características del módulo GSM/GPRS SIM 800L [28]

- ✓ Voltaje de operación: 3.4 hasta 4.4 V
- ✓ Consumo de corriente máximo: 500 mA
- ✓ Consumo de corriente (modo reposo) 0.7 mA
- ✓ Velocidad de transmisión GPRS 85.6 Kbps
- ✓ Protocolo TCP/IP en chip
- ✓ Soporta reloj en tiempo real (RTC)
- ✓ Tamaño de la SIM: Micro SIM
- ✓ Maneja cuatribanda

5.3.4.7 Ventajas del módulo GSM/GPRS SIM 800L [29]

Dentro de las ventajas podemos mencionar:

- ✓ Bajo consumo
- ✓ Opera cuatribanda
- ✓ Tamaño reducido, lo que lo hace ideal para integración con proyectos en PCB
- ✓ Bajo consumo en reposo: 0.7 Ma

5.3.4.8 Desventajas del módulo GSM/GPRS SIM 800L



Dentro de las desventajas que se pueden mencionar del módulo SIM800L se encuentran:

- ✓ Su hardware es poco desarrollado, para operaciones practicas es necesario el desarrollo de circuitos adicionales para su operación
- ✓ Posee menor alcance en comparación al SIM900L

5.3.4.9 Modulo GSM/GPRS elegido.

Para la elección del módulo GSM/GPRS se consideró el voltaje de operación, su hardware y tiempo de conexión entre otras propiedades de los módulos GSM/GPRS estudiados. En la tabla 8 se muestran algunas características tomadas en cuenta para la elección de este módulo dentro de este proyecto.

Tabla 8. comparación entre SIM900L Y SIM800L. Fuente: El autor.

GSM/GPRS's	SIM900L	SIM800L
prestaciones		
Voltaje de operación	9 a 20 V	3.6 a 4.6 V
Consumo de corriente	1.5 mA (2 A pico)	0.7 mA (2 A pico)
Hardware	desarrollado	Admite desarrollo
documentación	si	si
costo	40\$	18\$
tamaño	74x55 mm	25x23 mm

En la tabla 8 podemos notar un dato importante sobre el consumo de corriente y voltaje de operación; en el diseño del Sistema Electrónico para la Medición y Registro de Velocidad de un Vehículo Mediante GPS se debe considerar el óptimo funcionamiento

del conjunto completo, al tener la opción de operar con 9 a 20 voltios en el SIM900L y con un consumo pico de 2 A resulta más práctico que operar de 3.6 a 4.6 V con 2 A como lo hace el SIM800L, el hardware poco desarrollado del SIM800L requiere circuitería adicional que propone ruido en sus pines de entrada lo que genera problemas al momento de conectar a la red GSM, la disposición de una fuente de estas características requiere el uso de reguladores y disipadores de calor que volverían al sistema propenso a fallas. Este inconveniente se resuelve mediante el uso del SIM900L que puede operar sin ningún problema con fuentes que van desde 9 hasta los 20 V

5.3.5 Modulo micro SD

El adaptador usado para el almacenamiento de los datos obtenidos por el GPS. Pertenecer a la marca CATALEX en su versión V1.0. El módulo micro SD convierte el voltaje de 5 V a 3.3 V que es el voltaje de operación de la tarjeta SD. Cualquier voltaje superior 3.6 V dañara la tarjeta SD de ahí la necesidad de un módulo micro SD que convierta los niveles lógicos de voltaje. [30]

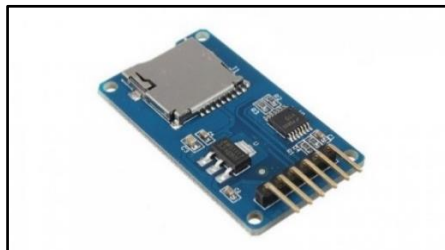


Figura 19. Modulo micro SD CATALEX. [30]

5.2.4.1 Características del módulo micro SD Catalex

- ✓ Admite tarjetas micro SD igual o mayor a 2 Gb
- ✓ Voltaje de entrada: 3.3 hasta 5.5 V
- ✓ Dimensiones 4.1x2.4 Cm
- ✓ 6 pines (GND, VCC, MISO, MOSI, SCK, CS). Donde: GND va a tierra, VCC a fuente de alimentación, MISO, MOSI, SCK van al bus SPI y CS es el pin de señal seleccionado por el chip

5.4 MATERIALES.

En la siguiente tabla se detallan los materiales empleados para el desarrollo del circuito para la medición y registro de velocidad mediante GPS

Tabla 9. Materiales empleados para la elaboración del circuito para la medición y registro de velocidad.
Fuente: El autor.

Materiales	Cantidad
SIM800L	1
SIM900L	1
PIC18F4620	1
LM7805	3
Arduino Mega	1
Modulo micro SD	1
SD Card 4G	1
GPS U-blox 6	1
LCD 16x2	1
Fuente de alimentación 4A	1
Ventilador 12 V DC	1

5.5 Diseño del sistema electrónico para la medición y registro de velocidad

El circuito principal se compondrá del conjunto de módulos debidamente seleccionados. Todo el sistema estará implementado según las especificaciones establecidas en los diferentes datasheets del fabricante, estos documentos se muestran en los anexos. En la figura 20 podemos ver el funcionamiento general de todo el Sistema de Medición y Registro de Velocidad Mediante GPS para un vehículo.

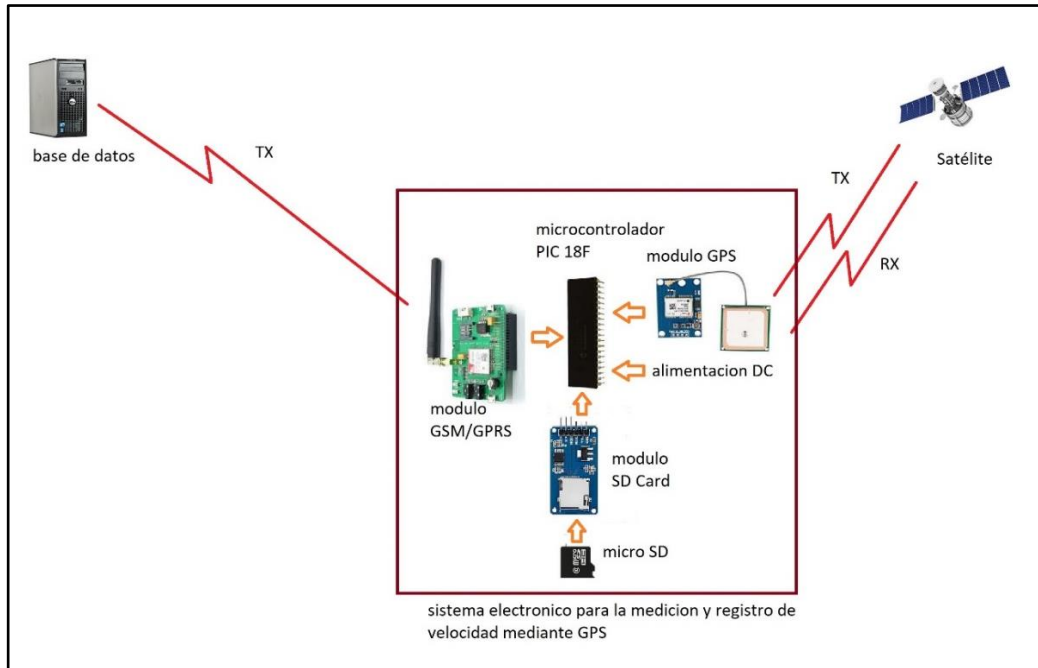


Figura 20. Funcionamiento del sistema para la medición y registro de velocidad. Fuente: El autor

Los módulos GSM/GPRS SIM 800L, GPS u-blox 6, SD card y pantalla LCD, se conectan al microcontrolador PIC 18F64620, este es el elemento encargado de procesar información,

Mediante una antena de microstrip el GPS establece comunicación con los satélites y estos responden enviando las tramas que contienen la información de ubicación, hora, fecha y velocidad que están en formato NMEA_2000.

Las tramas son procesadas de tal forma que sean legibles para el usuario, y se las muestra en una pantalla LCD. El microcontrolador también envía esta información a sus puertos a los que se encuentra conectado un módulo SD, en que se inserta una micro SD de 4Gb de capacidad de almacenamiento.

El microcontrolador además envía por su puerto serial al final del recorrido los datos que se almacenan en la micro SD a él módulo GSM/GPRS SIM 800L, y este lo envía a través de una antena hasta un receptor instalado en una PC, donde finalmente se almacenara el recorrido del vehículo.

5.6 Simulación del circuito

Para el desarrollo de este proyecto se usaron varios programas de los que se habla a continuación

5.6.1 Programas usados para la simulación

Después de entender la manera en la que se disponen los diferentes periféricos que componen este sistema, se procedió a simular para obtener resultados preliminares antes de ensamblar en protoboard y posteriormente en PCB. Las pruebas fueron realizadas mediante dos programas:

- ✓ **Proteus 7.8 Professional:** Este software se usó para la simulación dado que contiene las librerías necesarias para elementos de la familia de los PIC's, AVR, Arduino entre otros. Fue necesario la descarga de librerías de algunos módulos, en este caso será necesario descargar la librería para el uso del módulo GPS.
- ✓ **WinImage 9.0:** Este software se usó para la creación de imágenes de archivos de carácter extraíble. En este caso fue necesario crear una imagen virtual compatible con el formato FAT32.

En la figura 21 se muestra el circuito que se obtiene por medio del programa para la simulación. En este caso se usó el programa Proteus 7.8

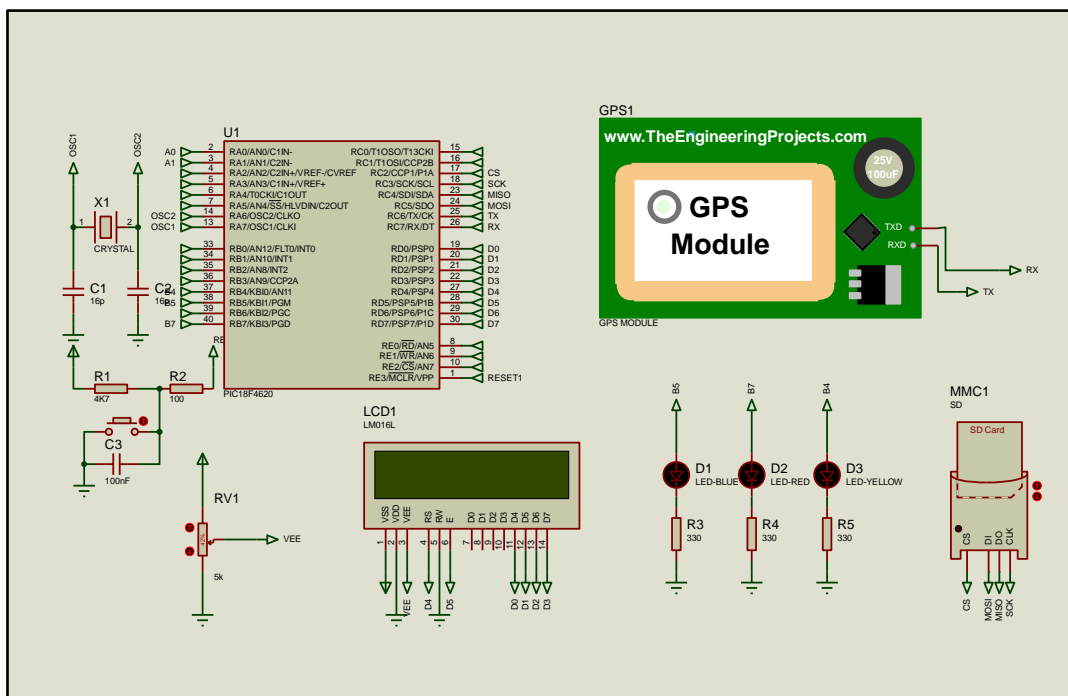


Figura 21. Circuito para la medición y registro de velocidad. Fuente: El autor.

En la figura 21 se muestra el circuito planteado para la medición y registro de velocidad. La funcionalidad de cada una de sus partes se detalla en el apartado 5.5.

5.6.2 Desarrollo de la simulación del circuito para la medición y registro de velocidad

El microcontrolador tiene como función recibir las tramas NME provenientes del módulo GPS. Los puertos analógicos han sido configurados como entradas digitales mediante la línea de código $ADCON1 = 0x0F$ ya que se está trabajando con módulos que presentan salidas digitales: como el módulo para SD card, la pantalla LCD y modulo GSM/GPRS. También se está usando una entrada serial del PIC 18F4620 mediante el puerto EUART, este puerto fue configurado a 9600 baudios, dado que el módulo GPS usa esta velocidad de transmisión.

El microcontrolador también tiene configurado 3 pines a los cuales se conectan 3 led indicadores. Estos representan el estado en que se encuentra el sistema. Esto se detalla a continuación.

1. Led correspondiente al pin B5 indica que la comunicación se ha establecido y se puede continuar con la ejecución del algoritmo. En la pantalla se muestra el mensaje “Inicio OK”
2. Led correspondiente al pin B7 indica que la tarjeta SD se encuentra insertada correctamente.
3. Led correspondiente al pin B4 indica que ha existido una falla en la comunicación. Esto se produce cuando el GPS no logra establecer comunicación con los 4 satélites que es requisito mínimo para empezar la comunicación entre GPS y los satélites y luego del GPS al PIC. Entonces se muestra el mensaje “Inicio fallido”

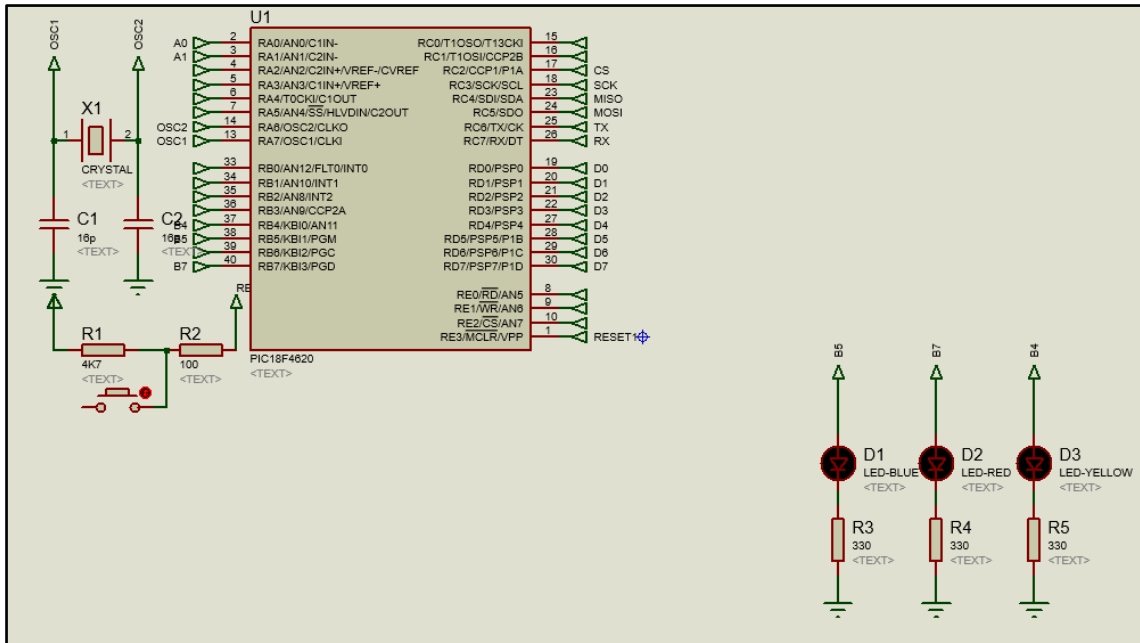


Figura 22. PIC18F4620 y LED indicadores. Fuente: El autor.

5.6.3 Integración del módulo GPS a la simulación.

El módulo GPS para este proyecto es el u-Blox que puede funcionar con una comunicación serial de 9600, 38400, 4800 y 57600. Es necesario para integrar este módulo al desarrollo del proyecto, descargar la librería correspondiente a este elemento que responde al nombre de GPS.HEX Esta librería se puede descargar de manera gratuita desde cualquier link de descarga. Para este proyecto se usa una comunicación de 9600 baudios que corresponde al formato NMEA-2000 y debe ser configurada en las propiedades de dicho elemento; que es el formato mejorado de NMEA-0183. Los datos transmitidos están en formato ASCII. Los caracteres entregados por el GPS son los siguientes.

- ✓ \$GPRMC
- ✓ \$GPGGA
- ✓ \$GPGSA
- ✓ \$GPGSV
- ✓ \$GPGLL
- ✓ \$GPVTG

En este proyecto se usa dos líneas de caracteres. La línea \$GPRMC, de la que se obtuvieron: hora, fecha, latitud y longitud. También se usa la línea de caracteres \$GPVTG, debido a que en esta línea de caracteres se puede obtener la velocidad en kilómetros por hora.

5.6.4 Configuración de la SD Card y ejecución de la simulación

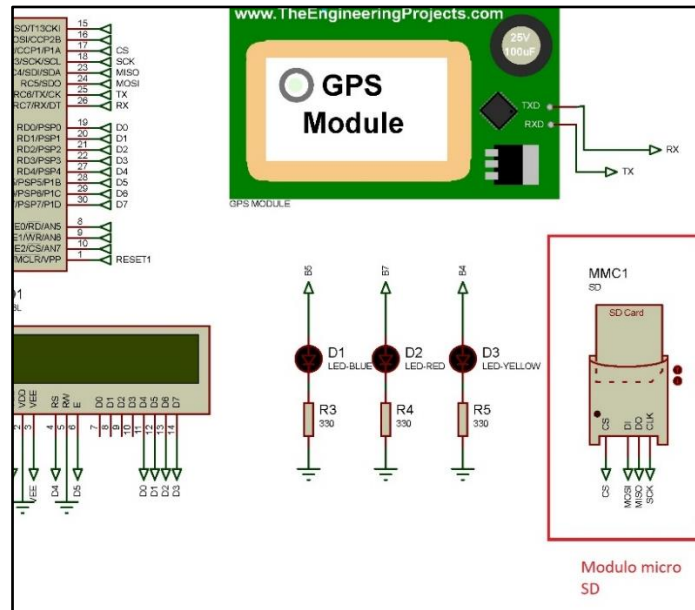


Figura 23. Módulo micro SD dentro del circuito. Fuente: El autor.

El módulo micro SD es el encargado de convertir los niveles de voltaje provenientes del PIC a los niveles que maneja la tarjeta SD este voltaje llega hasta 3.6 V. Dentro del PIC se manejó la librería correspondiente a FAT32, que es para tarjetas micro SD superiores a 2Gb. En este caso se está usando una tarjeta micro SD de 4Gb; aunque los archivos que se generan son únicamente *.txt. El PIC crea una carpeta llamada DIR_A en la que se genera un archivo de texto donde se guarda la información obtenida cada 5 segundos.

Luego de haberse creado el archivo de extensión *.txt en la pantalla LCD se muestra el mensaje “Archivos creados”. Posteriormente se empiezan a escribir los datos obtenidos desde el módulo GPS, en este momento se muestra en la pantalla el mensaje “Escribiendo...” y finalmente se visualiza el mensaje “Texto escrito correcto”, caso contrario se verá “Archivos no creados”, entonces se repetirá el proceso desde la obtención de tramas hasta escribir los datos sobre el archivo de texto creado al inicio de la función principal llamado DIR_A.

Para la simulación de la SD card es necesario ingresar algunos parámetros dentro de las propiedades dentro de la simulación. Para esto primeramente se debe crear los archivos con extensión *IMA. Para lograr este objetivo se usó el programa WinImage. El procedimiento se detalla a continuación:

1. Se abre el programa WinImage y creamos un nuevo documento

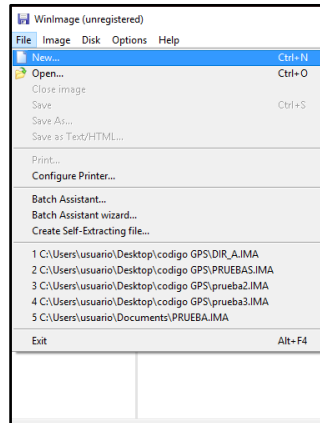


Figura 24. Se crea un nuevo documento. Fuente: El autor

2. Se selecciona *select custom image format* para ingresar parámetros de imagen avanzados y se da ok.

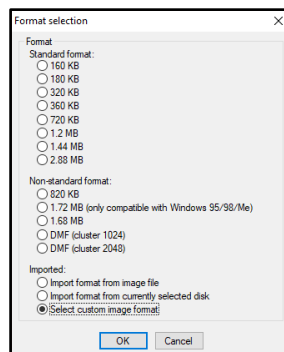


Figura 25. Selección de formato avanzado. Fuente: El autor

3. En *file sistem* se selecciona el formato de FAT32 para poder trabajar con micro SD superior a 2 Gb y se da ok.

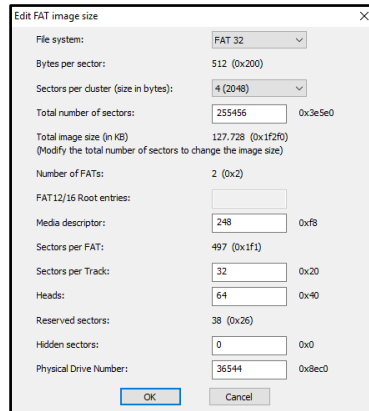


Figura 26. Se selecciona el formato FAT32. Fuente: El autor

4. Se desplaza a la parte de *save as..* y se selecciona la extensión del archivo como **IMA* y se adjunta el nombre de la carpeta que el programa en MikroC debe crear que para este caso es *DIR_A* y se guarda en cualquier carpeta. el nombre del archivo será *DIR_A.IMA*
5. Por último, se crear un documento de TIPO **txt* con el nombre del archivo que deberá crearse desde el PIC. Para este proyecto el archivo será *texto.txt* y se guarda en el mismo sitio donde se almaceno el archivo **IMA*

Ahora que se tiene los dos archivos creados, se editan las propiedades de la micro SD y se carga el archivo **IMA* y el archivo **txt*.

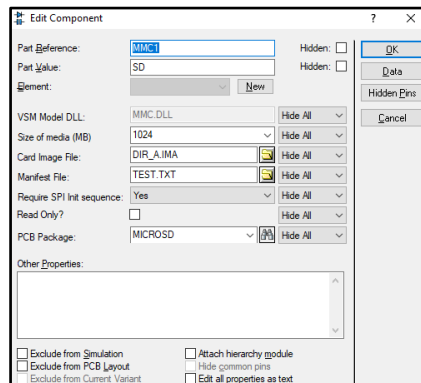


Figura 27. Se carga los archivos en edit properties. Fuente: El autor

Luego de cargar el archivo **HEX* generado por MikroC (que se describe en la sección 5.5) en el microcontrolador PIC18F4620, debemos cargar desde la librería el archivo **HEX* correspondiente al módulo GPS. En las especificaciones de la librería del módulo GPS, se especifica que la velocidad de transmisión que posee; que es de 9600 baudios y un cristal interno de 16Mhz

Después que se tienen los datos para la simulación de la micro SD, se procede a ejecutar la simulación. Primeramente, se obtendrá el mensaje de presentación configurado en el código en MikroC.

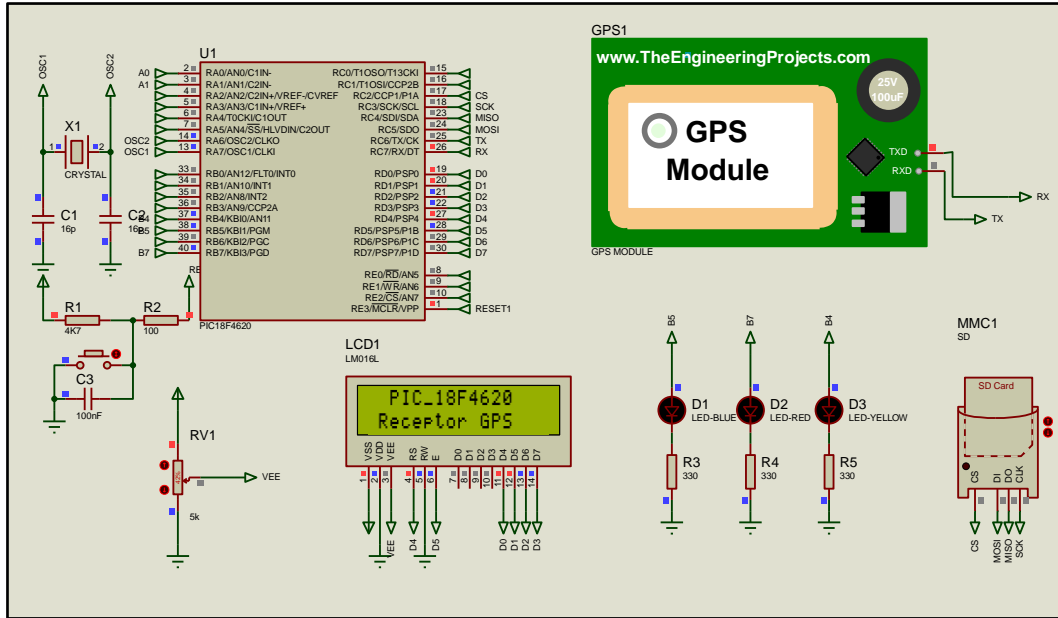


Figura 28. Mensaje de presentación obtenido desde la simulación en proteus 7.8. Fuente: El autor.

Posteriormente se visualizará que el dispositivo espera conexión, esto se necesita ya que el código debe esperar un tiempo hasta que el módulo GPS comience a recibir los datos desde los satélites que según el fabricante es de 4 minutos para el módulo U-blox 6.

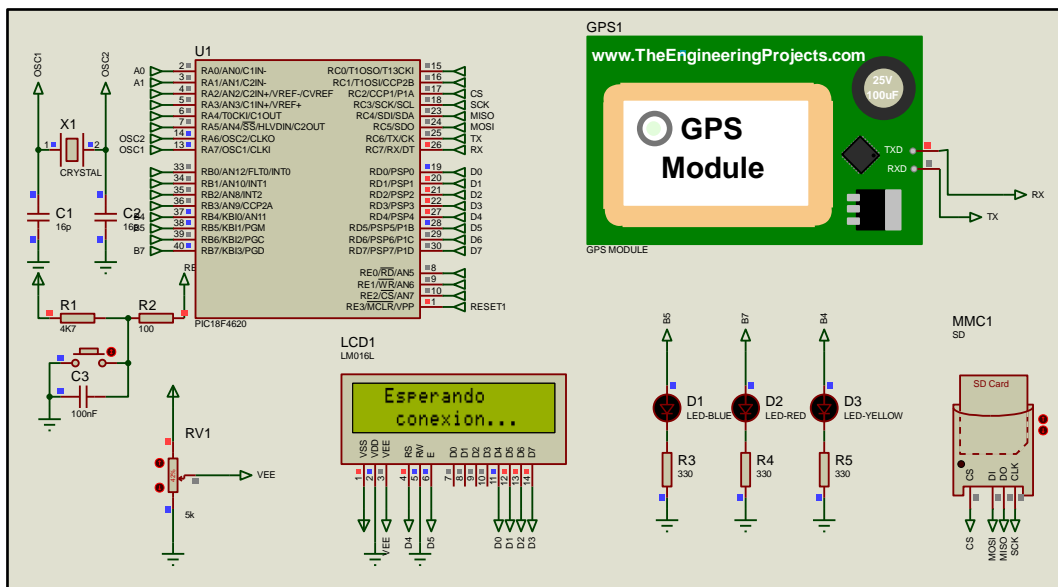


Figura 29. El dispositivo muestra el mensaje "Esperando conexión..." para dar tiempo que el módulo GPS establezca conexión. Fuente: El autor.

Durante este tiempo de conexión, también se verifica que la memoria micro SD se encuentre insertada correctamente. En la pantalla LCD se mostrará un mensaje de inicio seguro y encenderá el puerto B7. Se puede ver en la figura 30 el led de color rojo encendido y en la pantalla LCD el mensaje “Inicio OK”. Esto indica que la micro SD esta insertada correctamente.

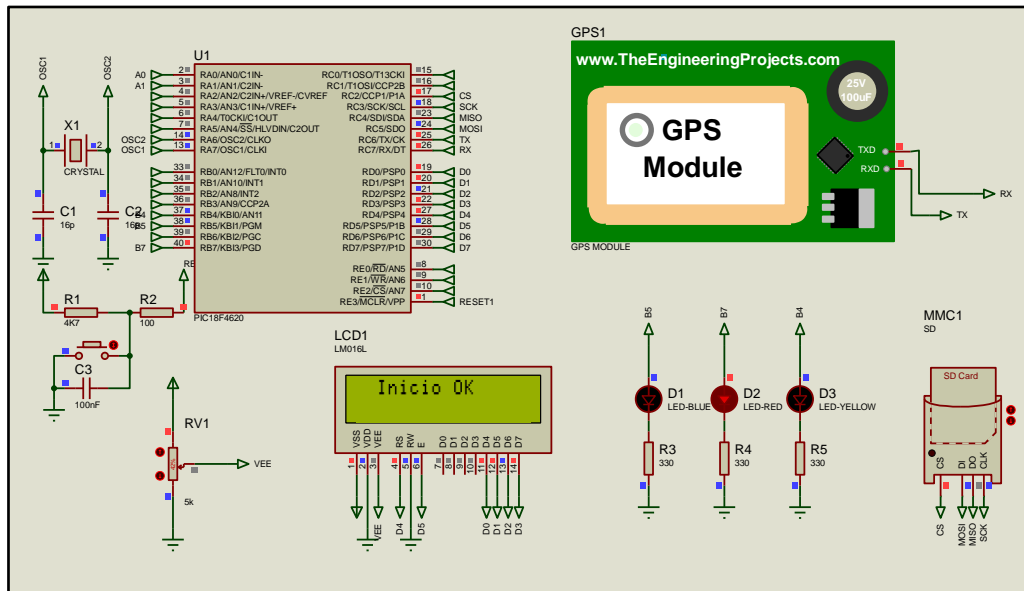


Figura 30. Si la micro SD se encuentra insertada correctamente se muestra "Inicio OK" y enciende el puerto B7.
Fuente: El autor.

En el caso que la micro SD no se encuentre insertada correctamente se encenderá el puerto B4 y mostrara el mensaje “Inicio fallido”.

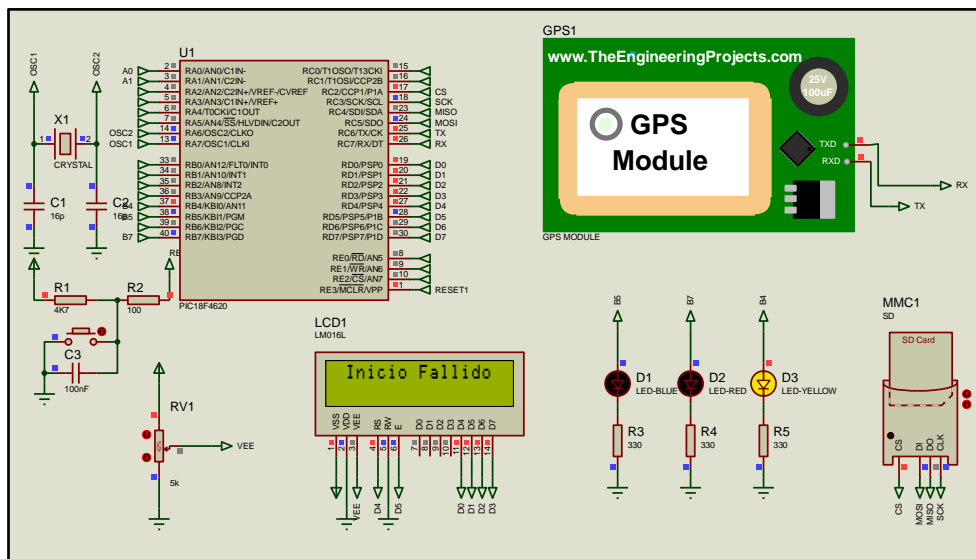


Figura 31. Si la micro SD no está insertada correctamente se muestra el mensaje "inicio fallido" y se enciende el led B4. fuente: El autor.

Si la micro SD se encuentra insertada correctamente, entonces el código comprueba la conexión del módulo GPS y procede a crear la carpeta en la que se guardara el archivo de texto.

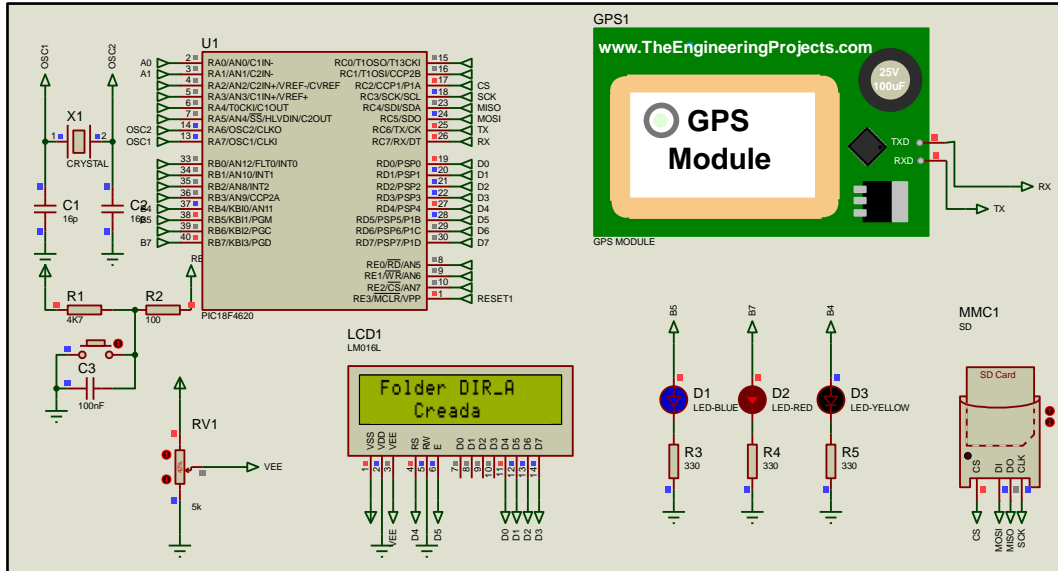


Figura 32. El código crea la carpeta DIR_A en la que se guardaran los datos adquiridos. Fuente: El autor.

Se puede notar que el LED del puerto B5 se enciende, lo que indica que el GPS está recibiendo datos provenientes de los respectivos satélites.

Luego de esto se podrá visualizar en la pantalla LCD los datos obtenidos por el módulo GPS. Primeramente, mostrara fecha y hora, luego se visualiza la ubicación.

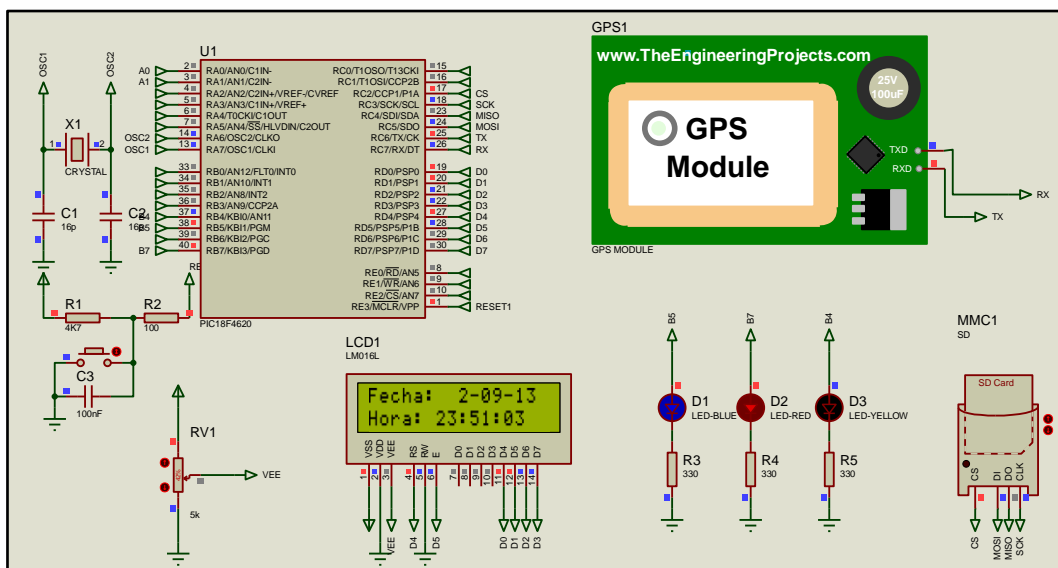


Figura 33. El GPS recibe los datos de los satélites y pueden visualizarse en la pantalla LCD. Fuente: El autor.

También se ve en la pantalla LCD los valores correspondientes de latitud y longitud.

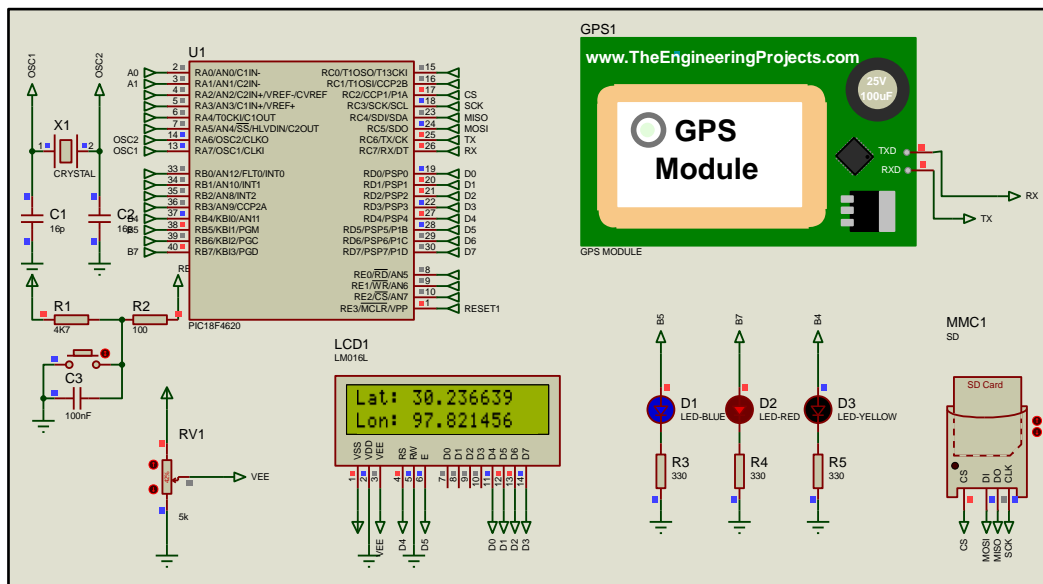


Figura 34. Valores de latitud y longitud durante la simulación. Fuente: El autor.

Es importante mencionar que durante la simulación no es posible obtener valores de velocidad, esto se debe a que el código analiza las tramas GPVTG para obtener velocidad y la trama GPRMC para hora, fecha y posición. Mientras que analizando las tramas del módulo GPS a través del terminal virtual de proteus, se puede ver que no se tiene la trama GPVTG.

```

Virtual Terminal
$PGRGA,045252.000,3014.4273,N,09749.0628,W,1,09,1.3,206.9,M,-22.5,M,,0000*6F
$GPRMC,045103.000,A,3014.1984,N,09749.2872,W,0.67,161.46,030913,,A*7C
$PGRGA,045104.000,3014.1985,N,09749.2873,W,1,09,1.2,211.6,M,-22.5,M,,0000*62
$GPRMC,045200.000,A,3014.3820,N,09748.9514,W,36.88,65.02,030913,,A*77
$PGRGA,045201.000,3014.3864,N,09748.9411,W,1,10,1.2,200.8,M,-22.5,M,,0000*6C
$GPRMC,045251.000,A,3014.4275,N,09749.0626,W,0.51,217.94,030913,,A*7D
$PGRGA,045252.000,3014.4273,N,09749.0628,W,1,09,1.3,206.9,M,-22.5,M,,0000*6F
$GPRMC,045103.000,A,3014.1984,N,09749.2872,W,0.67,161.46,030913,,A*7C
$PGRGA,045104.000,3014.1985,N,09749.2873,W,1,09,1.2,211.6,M,-22.5,M,,0000*62
$GPRMC,045200.000,A,3014.3820,N,09748.9514,W,36.88,65.02,030913,,A*77
$PGRGA,045201.000,3014.3864,N,09748.9411,W,1,10,1.2,200.8,M,-22.5,M,,0000*6C
$GPRMC,045251.000,A,3014.4275,N,09749.0626,W,0.51,217.94,030913,,A*7D
$PGRGA,045252.000,3014.4273,N,09749.0628,W,1,09,1.3,206.9,M,-22.5,M,,0000*6F
$GPRMC,045103.000,A,3014.1984,N,09749.2872,W,0.67,161.46,030913,,A*7C
$PGRGA,045104.000,3014.1985,N,09749.2873,W,1,09,1.2,211.6,M,-22.5,M,,0000*62
$GPRMC,045200.000,A,3014.3820,N,09748.9514,W,36.88,65.02,030913,,A*77
$PGRGA,045201.000,3014.3864,N,09748.9411,W,1,10,1.2,200.8,M,-22.5,M,,0000*6C
$GPRMC,045251.000,A,3014.4275,N,09749.0626,W,0.51,217.94,030913,,A*7D
$PGRGA,045252.000,3014.4273,N,09749.0628,W,1,09,1.3,206.9,M,-22.5,M,,0000*6F
    
```

Figura 35. Tramas del módulo GPS simulado en proteus obtenidas desde el Virtual Terminal. Fuente: El autor

El módulo GPS u-blox si presenta las tramas GPVTG por lo que en la práctica no se presenta este inconveniente.

Después de que el código muestra en la pantalla LCD los datos obtenidos por el módulo GPS. El código indica que los archivos han sido creados correctamente, entonces se muestra en la pantalla “Archivos creados”.

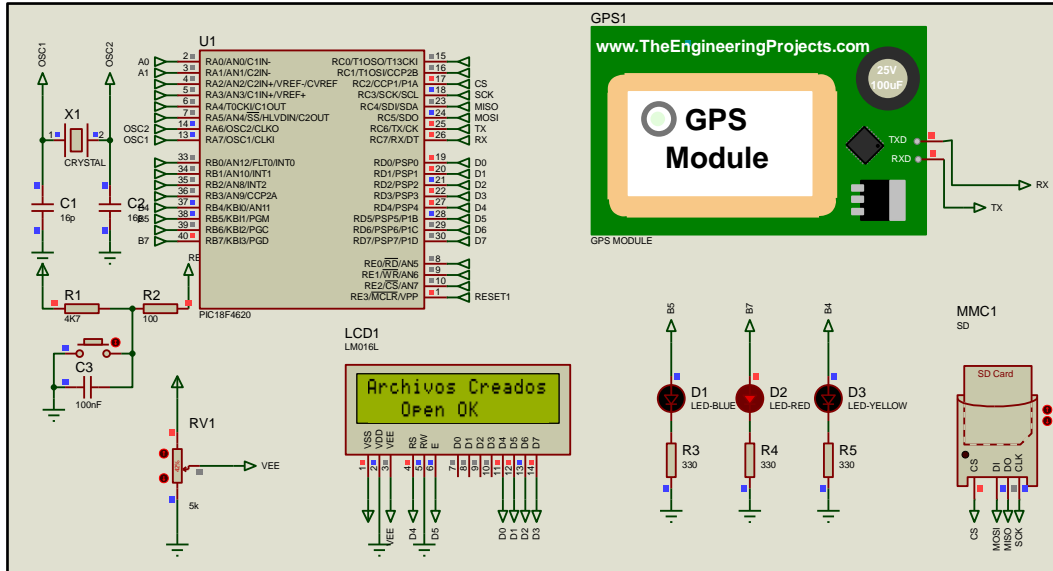


Figura 36. El código indica que los archivos han sido creados correctamente. Fuente: El autor.

La parte siguiente del código ahora indica que los datos están siendo escritos en la micro SD y muestra el mensaje “Escribiendo...”.

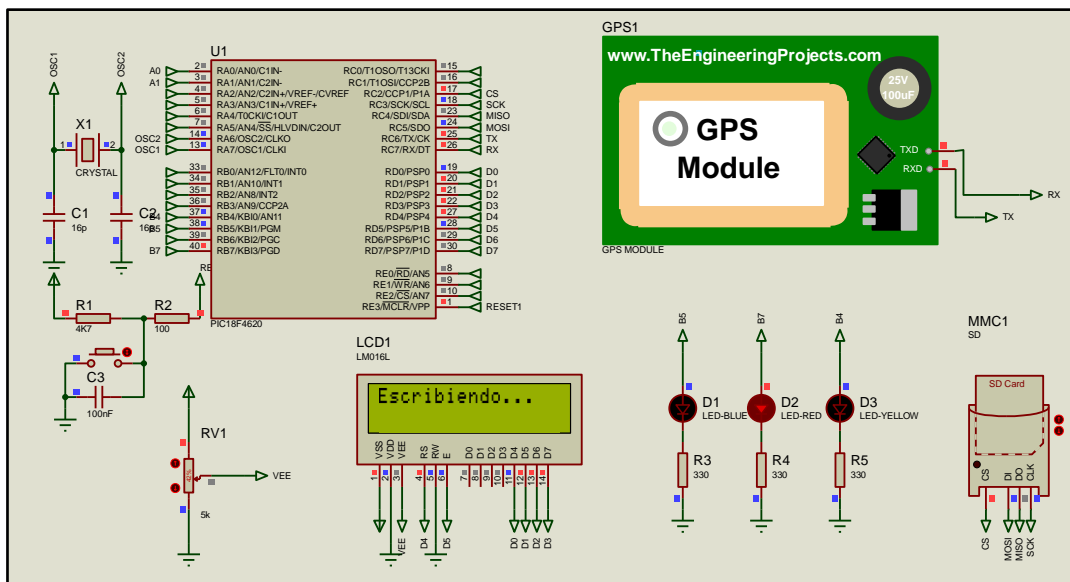


Figura 37. El código indica por medio de la pantalla LCD que los datos se están escribiendo en la micro SD. Fuente: El autor.

Finalmente se indica en la pantalla LCD que los datos obtenidos por el módulo GPS han sido almacenados en la micro SD, mostrando el mensaje “texto escrito correcto”, luego

de esto el sistema vuelve a indicar los datos que se están obteniendo por el módulo GPS de la figura 33 y 34.

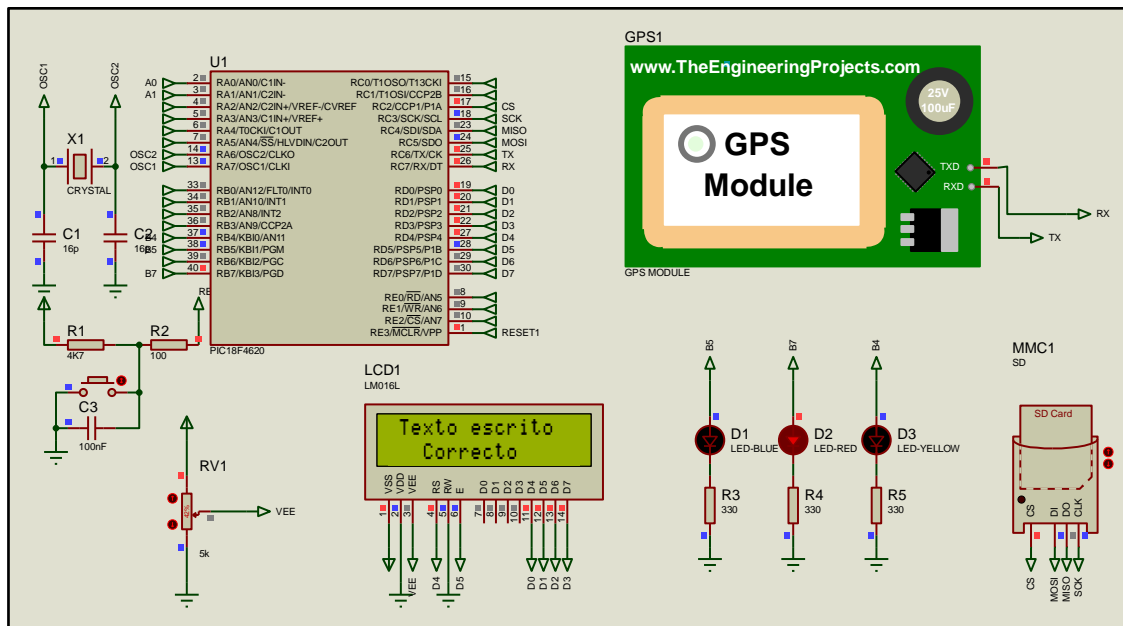


Figura 38. El mensaje "texto escrito correcto" indica que los datos obtenidos por el módulo GPS han sido almacenados en la micro SD. Fuente: El autor.

Para acceder a los datos almacenados en la micro SD se cierra la simulación y se debe abrir el programa WinImage y luego se abre el archivo que se creó para almacenar el texto generado por el código para el PIC. En este caso se tiene que buscar el archivo DIR_A.IMA, luego de abrir se extraer la información que se ha guardado en esta carpeta.

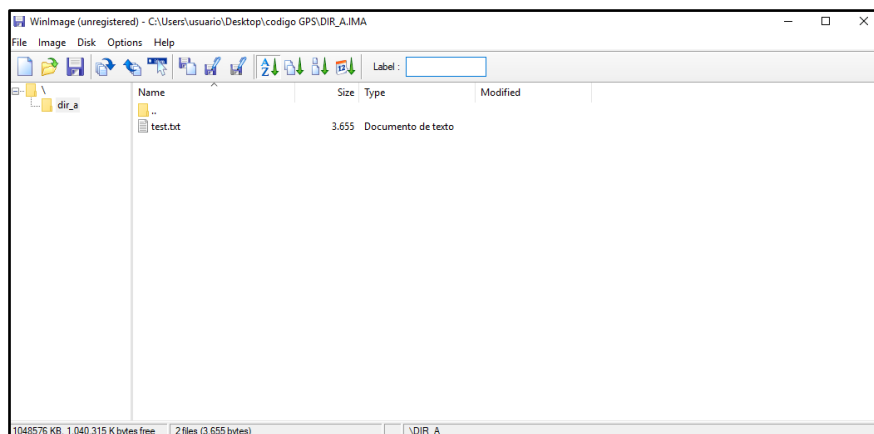


Figura 39. Archivo creado en WinImage. Fuente: El autor.

El documento es un archivo de tipo texto con el nombre que se especificó en el código en MikroC. En la práctica estos datos se almacenarán cada 5 segundos.

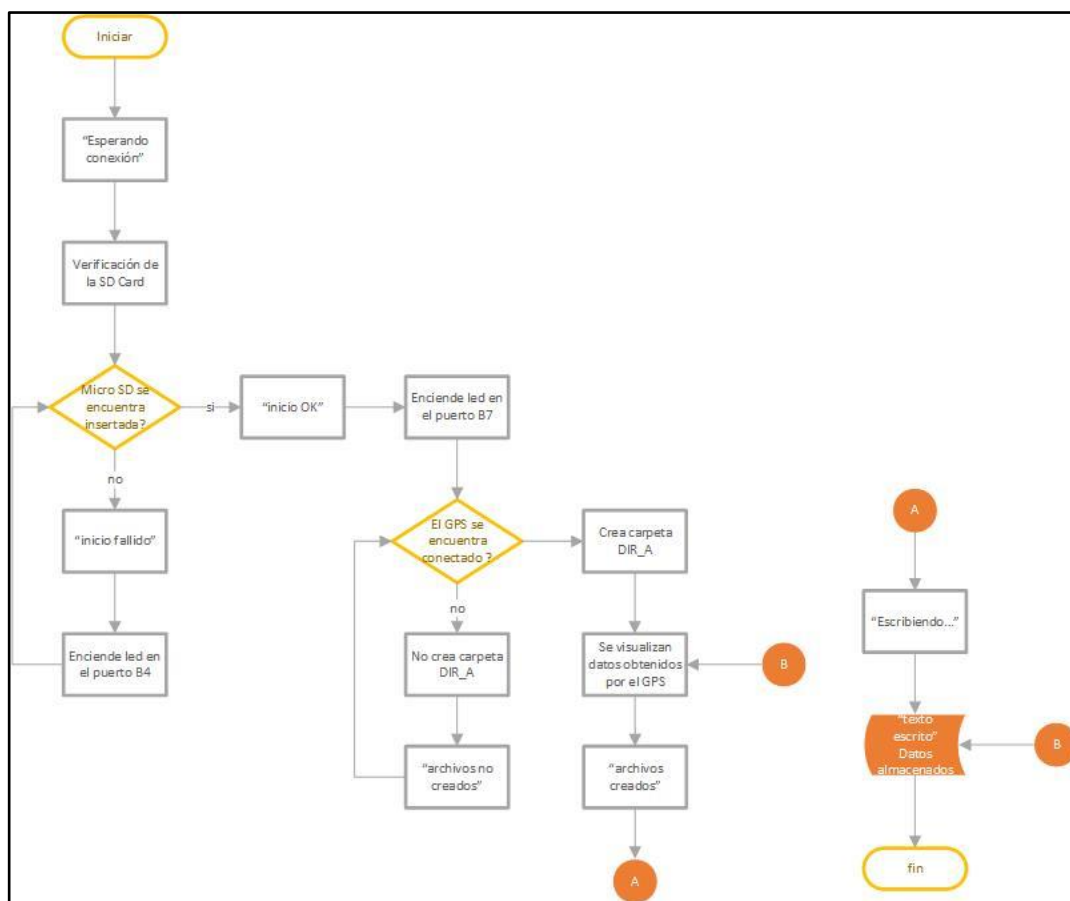


Figura 41. Diagrama de flujo del código para el microcontrolador. Fuente: El autor

El sistema de medición y registro de velocidad maneja una Pantalla LCD de 16x2 caracteres, para que de esta forma el usuario sepa los procesos que se están ejecutando internamente. El algoritmo se describe en la a continuación y el código desarrollado en MikroC se encuentra en los anexos

1. Se inicializa el sistema
2. Se espera un tiempo de 5 segundos
3. Se verifica que la SD Card se encuentre en su posición correcta para recibir datos
 - 3.1 si no está insertada podremos visualizar en la pantalla “inicio fallido”
 - 3.2 Se debe insertar la SD Card y reiniciar el sistema
 - 3.3 De estar insertada correctamente se visualiza en la LCD “inicio OK” y enciende un led de color azul conectado al puerto B7
4. Se verifica que el módulo GPS se encuentra conectado correctamente
 - 4.1 Si el módulo GPS no está conectado correctamente no crea la carpeta en la que se almacenara el archivo de texto.

- 4.2 Luego vuelve a verificar que el módulo GPS esté conectado correctamente
- 4.3 De estar correctamente conectado el módulo GPS. Entonces se crea la carpeta en la que se almacenara los datos que se han obtenidos desde las tramas que transmite el GPS al PIC por medio de el puerto serial UART.
- 4.4 Luego se visualiza que en la pantalla aparece la frase “archivos creados”
- 4.5 Posteriormente estos archivos son escritos en la SD Card y se verá en la pantalla “escribiendo...” que significa que los datos están siendo escritos en la carpeta creada que para este proyecto tendrá en nombre de DIR_A
5. Se muestra un mensaje en el que se muestra que los archivos ya han sido escritos correctamente y que se ha almacenado en la SD Card, este mensaje dice “texto escrito correcto”
6. Luego se vuelve al apartado 4

Nota: Se puede notar que no existe ninguna restricción en caso que el módulo GPS no esté recibiendo datos desde los satélites, esto se debe a que durante los primeros minutos el módulo GPS aún debe establecer la ruta por la que se conectara a los satélites existentes durante el intervalo de tiempo que esté conectado.

5.8 código para visualización de los datos obtenidos.

Para la visualización en el ordenador, fue necesario desarrollar un código que permita visualizar y generar un archivo de texto que tiene como función almacenar los datos obtenidos por el GPS, este código permite la conexión mediante puerto serial para analizar los datos recibidos por el SIM800L que funciona como elemento receptor.

5.8.1 Software para la elaboración del código para la visualización de los datos obtenidos.

Dentro de los programas que se consideraron para la elaboración del código que permita visualizar los datos obtenidos por el GPS, se analizaron varias opciones, entre las cuales tenemos las siguientes:

- ✓ Python 3.7: Se considero el uso de este programa ya que sus requerimientos de hardware no presentan mayores dificultades al momento de la elaboración de proyectos, cuenta con gran versatilidad de librerías que se pueden usar libremente.
- ✓ Matlab 2018: Dentro de las características más importantes que posee este software esta la función del Simulink que permite el desarrollo de códigos mediante

programación por bloques, pero con la limitante que las librerías dentro de esta función son bastante limitadas.

- ✓ LabView 2019: cuenta con múltiples librerías que permiten acceder a puertos seriales y manejar información a partir de este, su característica principal se basa en la programación gráfica, mediante el uso de bloques que se organizan de manera que cumplan una función específica

Tabla 10. Comparación de las principales características de Python, Matlab y Labview. Fuente: El autor.

Software Características	Python 3.7	Matlab 2018	LabView
Librerías	Librerías de difícil acceso	Librerías de difícil acceso	Existe gran cantidad de librerías disponibles
Documentación	Existe buena documentación	Existe poca documentación sobre el desarrollo de proyectos	Existe buena documentación sobre desarrollo de proyectos con este software
Interfaz	La programación de interfaces es complicada	Cuenta con una Guide que permite desarrollar interfaces	Permite el desarrollo de código e interfaz de manera simultánea
Requerimientos de hardware	Muy bajos	Muy altos	Muy altos

Por lo tanto, según la información antes mencionada y según los requerimientos que maneja este proyecto como; visualizar datos mediante puerto serial y almacenar esta información, se optó por el uso del software LabView 2019, ya que sus librerías se adaptan a las necesidades de este proyecto, además, existe gran cantidad de información sobre desarrollo de proyectos basados en el uso de puertos seriales

5.8.2 Desarrollo del código para la visualización y almacenamiento de los datos obtenidos

En la figura 42 se puede visualizar el bloque encargado del acceso al puerto serial y las configuraciones de velocidad necesarias para la comunicación, en este caso se trabajó con 9600 Baudios. En este puerto se ingresa el COM al que está conectado el receptor, en este caso un SIM800L.

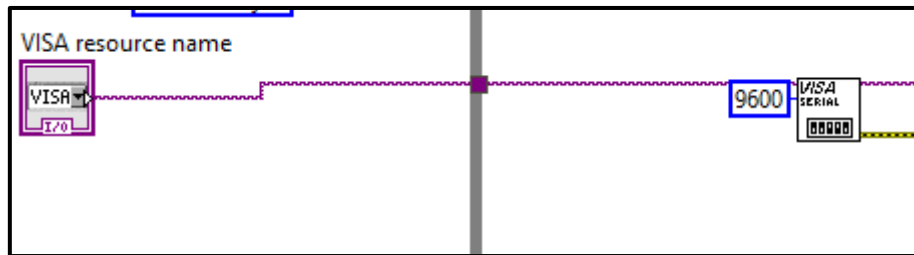


Figura 42. Bloque para acceder al puerto serial. Fuente: El autor.

Luego es necesario tener una lectura y escritura del puerto serial, lo que se logra con los bloques de la figura 43. La escritura corresponde a los comandos AT que verifican los mensajes en la memoria del SIM800L.

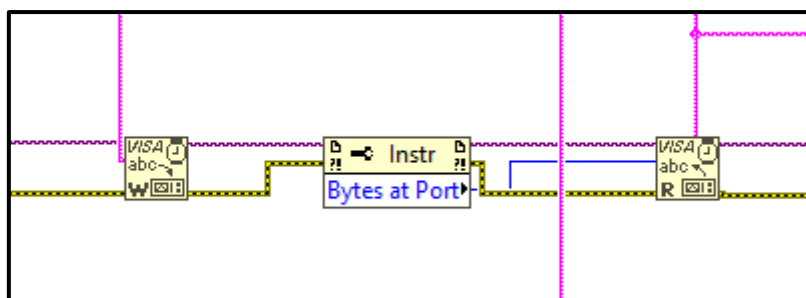


Figura 43. Lectura y escritura del puerto serial. Fuente: El autor.

Dentro del ciclo de lectura y escritura del puerto serial, es necesario finalizar dicho proceso mediante el bloque Close VISA para así evitar conflictos y errores al momento de reconocer el puerto cuando se ejecutó el código. Esto se evidencia en la figura 44.

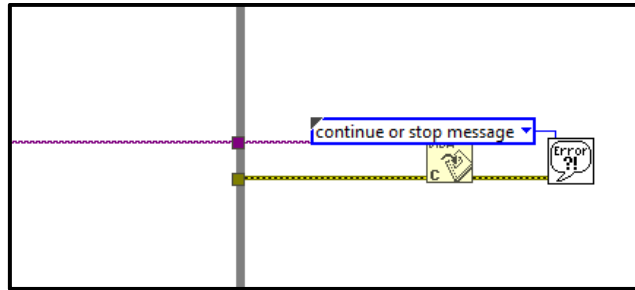


Figura 44. Bloque encargado de cerrar el puerto serial. Fuente: El autor.

En la figura 45, se muestra el bloque llamado File Dialog, este bloque se encarga de crear el directorio en el que se guardarán los datos que se leen en el puerto serial, se debe especificar un nuevo nombre siempre que se ejecute el código.

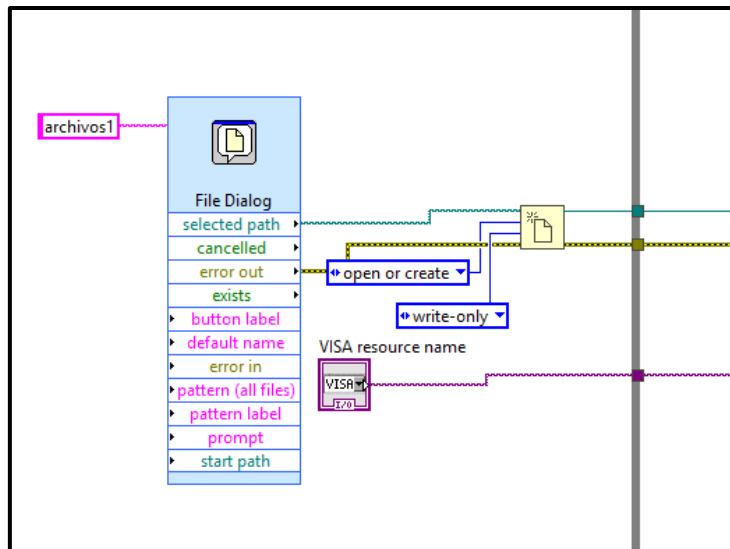


Figura 45. bloque encargado de crear directorio y guardar datos. Fuente: El autor.

Luego, para obtener datos decimales de valores como latitud y longitud o velocidad, y poder ser graficados o usados para operaciones, es necesario realizar una conversión. Previo a esto se realiza un filtrado de string, ya que toda la lectura se la hace desde datos de este tipo. En la figura 46 se muestran los bloques usados para el filtrado de string y posteriormente la conversión de estos datos a valores decimales.

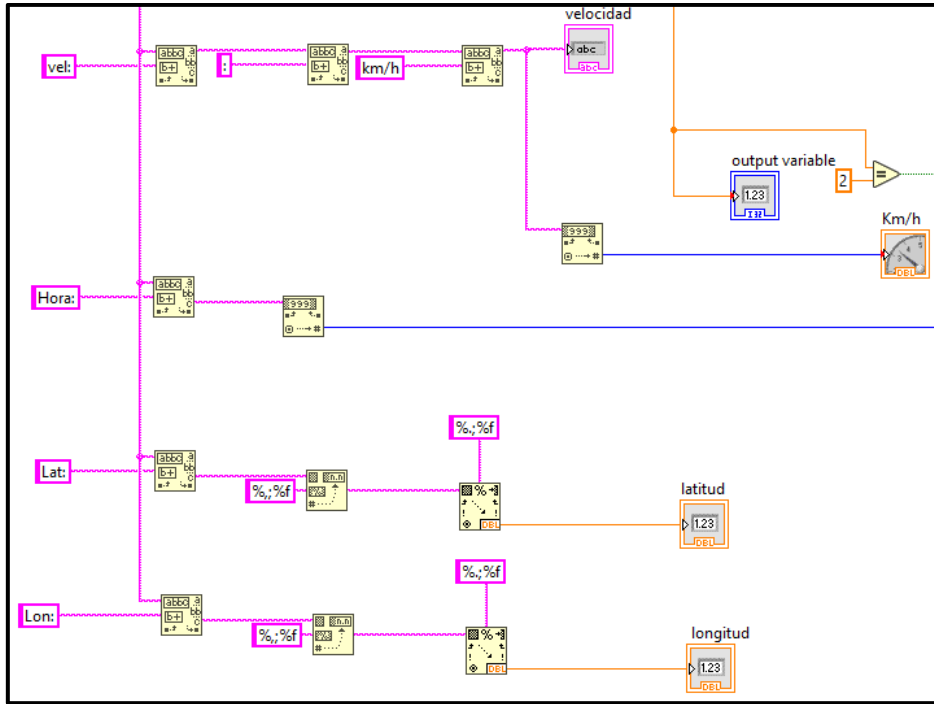


Figura 46. Filtrado de String y conversión a dato numérico. Fuente: El autor.

Además de los bloques ya mencionados se agregaron algunas funciones para detener la simulación, graficar entre otras. Dadas las dimensiones del código completo, se lo muestra en los Anexos 2

En la figura 47 se puede ver el panel de control del código generado por el diagrama de bloques que contiene el código para la visualización y almacenamiento de los datos obtenidos por el GPS.

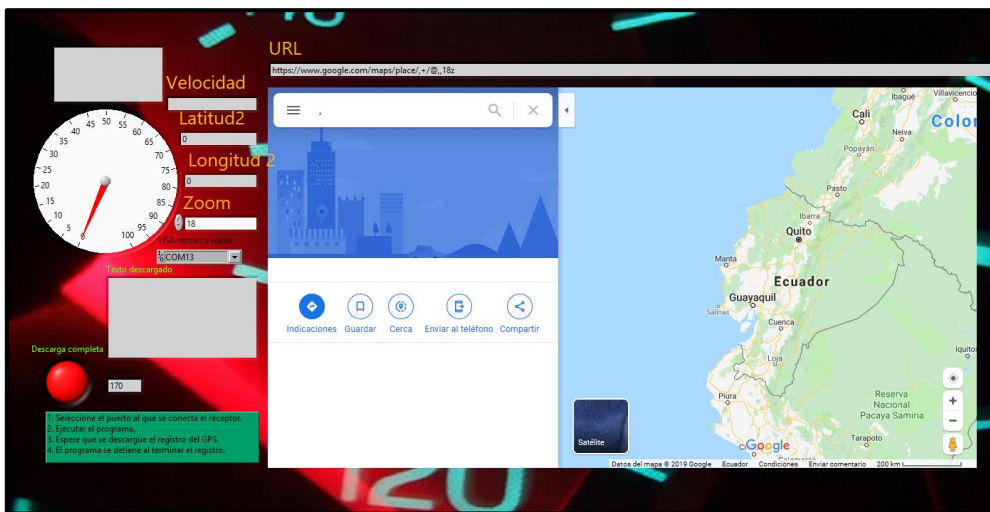


Figura 47. panel de control del VI generado. Fuente: El autor.

5.8.3 Desarrollo de plataforma de posicionamiento.

Para obtener una visión más precisa del punto en el que se encuentre un vehículo en el que se haya instalado el circuito para la Medición y Registro de Velocidad Mediante GPS. Es necesario poder visualizar estos datos a través de una plataforma de posicionamiento y así tener una imagen de las coordenadas enviadas por el SIM900L.

La herramienta de Google Maps permite incorporar mapas de manera gratuita en proyectos que operan con Hardware o software libre. Dentro de las características importantes que posee el software Labview, está la librería Gmaps que permite operar con mapas de Google y ubicar un determinado punto, permite la opción de trabajar con URLs y también permite trabajar con códigos en HTML. La opción más practica para el desarrollo de este proyecto se basó en el uso de URLs ya que de esta forma se puede visualizar mapas sin necesidad de acceder a ninguna cuenta o claves API (Application Programming Interface).

En la figura 48 se puede ver el código incorporado para acceder a los mapas de Google Maps en el cual no fue necesario el uso de alguna librería especial.

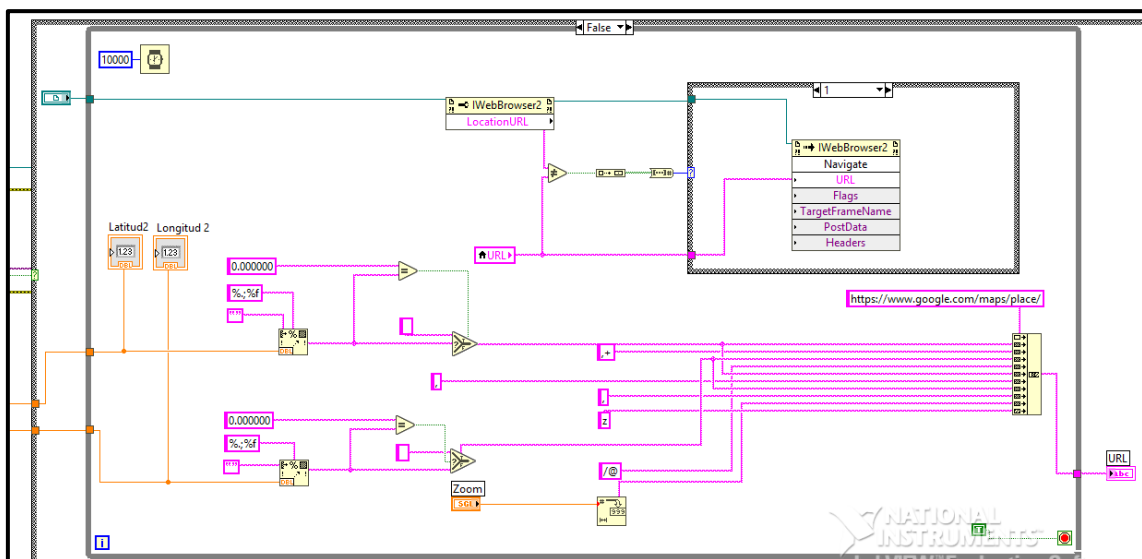


Figura 48. Código para la plataforma de posicionamiento. Fuente: El autor.

Se puede explicar el funcionamiento del código para la plataforma de posicionamiento de la siguiente manera:

- ✓ Se establece una dirección web, mediante el uso del bloque Web Browser.

- ✓ Se crean las variables locales para poder llamar esta URL en los sitios requeridos, como por ejemplo la pantalla.
- ✓ Se concatena la información que forma la URL que tiene como página principal <https://www.google.com/maps/place/>
- ✓ Se usan los datos de latitud y longitud convertidos en datos decimales para poder trabajar con el bloque de concatenar de Labview

6. RESULTADOS

Luego de analizados los datos obtenidos a través de la simulación se procede a realizar las pruebas en una placa PCB. Estas pruebas fueron previas a la placa final, en la que se puede visualizar los resultados esperados según la simulación.

6.1 Resultados en el PCB para pruebas

En la figura 49 se puede observar los distintos módulos usados para el desarrollo de este proyecto, en la parte superior izquierda se puede ver el módulo GPS u-blox 6, en la parte inferior podemos observar el módulo micro SD Card, en la pantalla se puede visualizar los datos de latitud y longitud que permanecen cero, lo que indica que el módulo GPS aun no recibe la información de posición de los satélites.

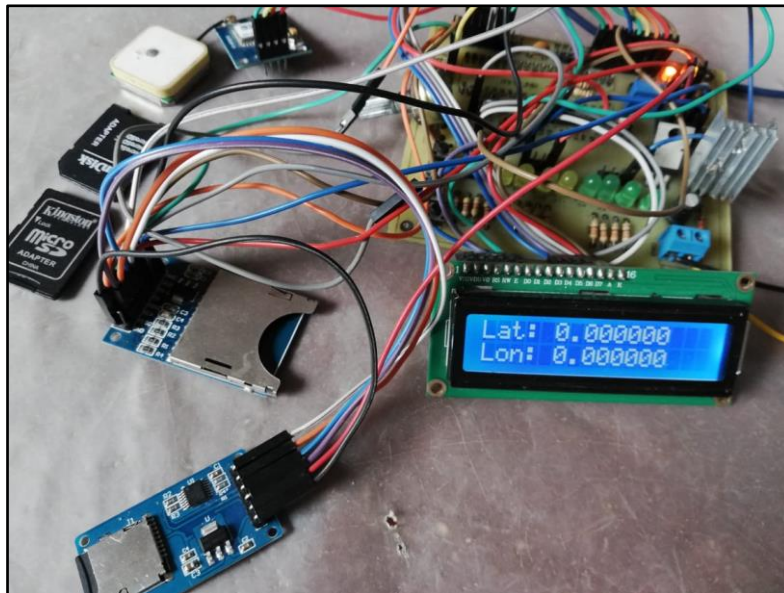


Figura 49. PCB para pruebas. Fuente: El autor.

En la figura 50 se puede observar el mensaje de inicio configurado en el microcontrolador “PIC_18F4620 Receptor GPS”, este mensaje da tiempo a que se establezcan las conexiones entre el módulo GPS y los respectivos satélites.



Figura 50. mensaje de presentación. Fuente: El autor.

En la figura 51 se puede verificar el mensaje de “Esperando conexión” este mensaje también tiene como propósito guardar tiempo para que se establezca conexión, así como verificar que la micro SD Card se encuentre insertada correctamente.



Figura 51. mensaje de espera en el que se verifica el estado de la micro SD Card. Fuente: El autor.

En el caso que se desconecte el dispositivo GPS o no se reciban datos dentro del tiempo configurado en el código, no se crearan los datos y la pantalla mostrara el texto de la figura 52.



Figura 52. Este mensaje se muestra si se desconecta el módulo GPS o se supera el tiempo de espera para establecer conexión GPS/satélites. Fuente: El autor.

Si el sistema empieza a recibir información proveniente del GPS, se visualiza el mensaje de la figura 53 que indica los primeros datos de coordenadas, fecha y velocidad.



Figura 53. Coordenadas obtenidas por el GPS. Fuente: El autor.



Figura 54. Velocidad obtenida por el módulo GPS con un vehículo estático. Fuente: El autor.



Figura 55. datos de fecha y hora obtenidos por el módulo GPS. Fuente: El autor.

Luego, estos datos serán dirigidos a la carpeta DIR_A que se crea para almacenarlos en la micro SD Card, este paso se verifica con el mensaje que se muestra en la figura 56.

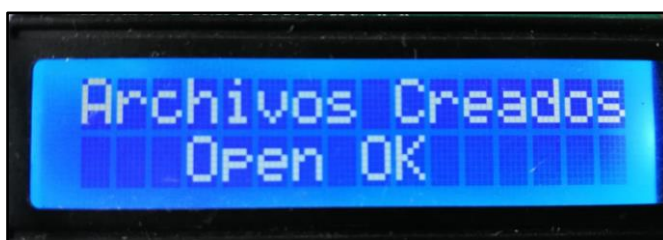


Figura 56. Se presenta el mensaje "Archivos creados" que indica que la carpeta en la que se guardarán los archivos ha sido creada y el GPS está en conexión. Fuente: El autor.

A continuación, estos datos obtenidos se escriben en la micro SD Card y se muestra el mensaje de la figura 57.

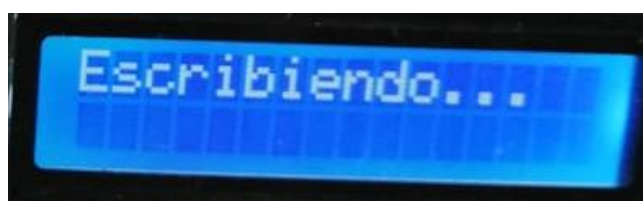


Figura 57. mientras el sistema escribe los archivos en la micro SD Card se muestra "escribiendo". Fuente: El autor.

Finalmente se muestra un mensaje que indica que todo el proceso se ha cumplido con normalidad, esto se ve en la figura 58.



Figura 58. "Texto escrito correcto" indica q todo el proceso de almacenamiento se ha desarrollado con normalidad. Fuente: El autor.

6.2 Resultados del PCB final

En la figura 59 se puede visualizar el circuito PCB final para el procesamiento de los datos obtenidos por el GPS u blox-6. El circuito consta de un elemento principal que es el PIC 18F460, circuitería adicional propuesta en el datasheets para su operación, leds indicadores de los que se habló en la sección 5.6.2 y pines adicionales para su programación mediante el puerto serial USART. Los datos obtenidos por el GPS, son procesados por el microcontrolador PIC18F4620, almacenados en la micro SD Card de 4G y enviados a través del módulo SIM900L.

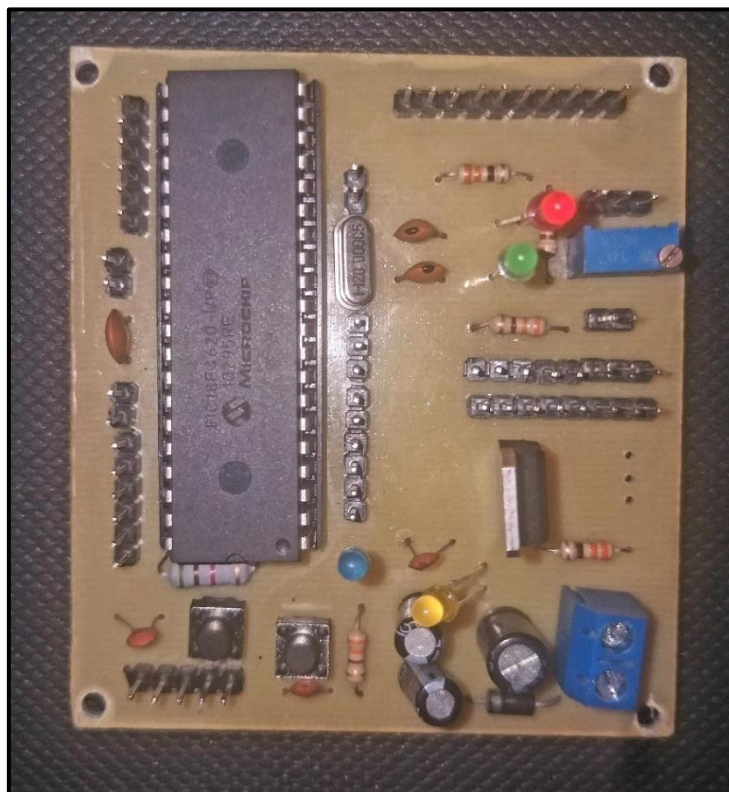


Figura 59. PCB para el procesamiento de datos obtenidos desde GPS u blox-6. Fuente: El autor

En la figura 60 se puede ver el sistema total armado, donde se puede evidenciar que los diferentes módulos se encuentran conectados de la manera que se indicó en el apartado 5.5 (Diseño del Sistema Electrónico para la Medición y Registro de Velocidad).

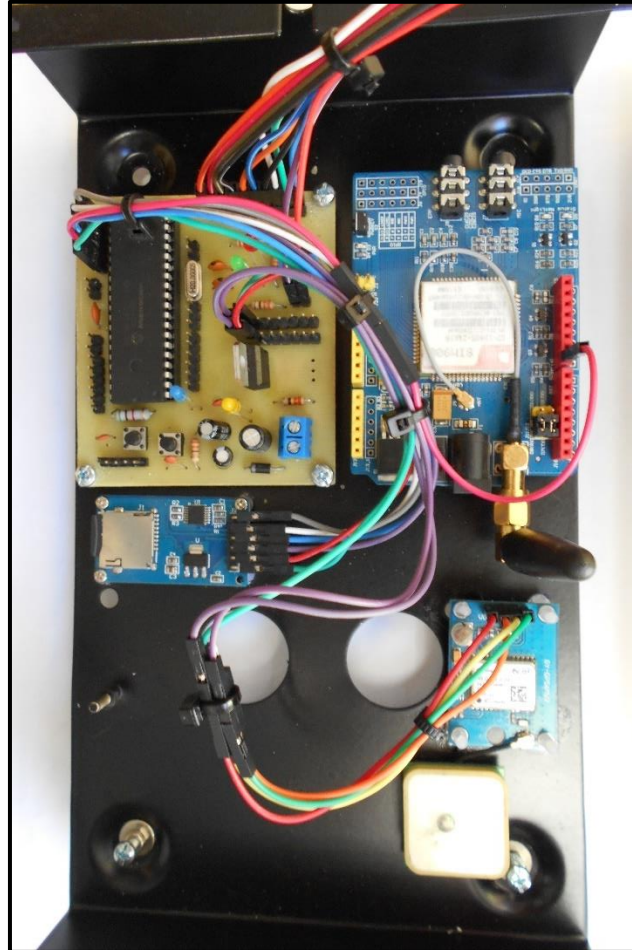


Figura 60. Sistema completo para la medición y registro de velocidad mediante GPS. Fuente: El autor.

En la figura 60 se puede notar la forma definitiva en la que los diferentes módulos quedan dispuestos, los diferentes módulos se encuentran respectivamente aislados de su superficie y se encuentran ubicados en una caja metálica en la operaran de manera permanente.

Además, los resultados obtenidos por el GPS u blox-6 se muestran en su tapa frontal también metálica en una pantalla LCD de 16x2 caracteres, fijados con tornillos milimétricos, esto se muestra en la figura 61.

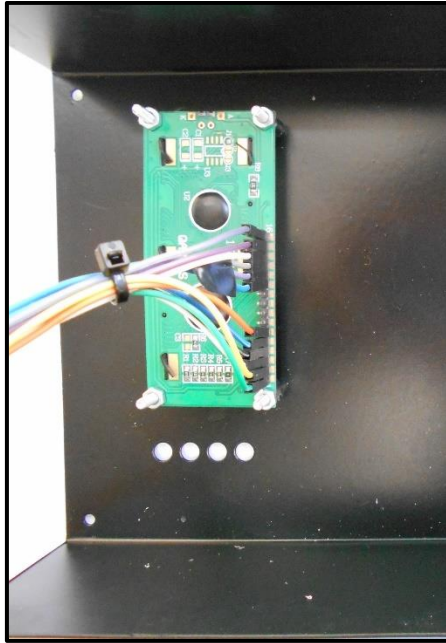


Figura 61 pantalla LCD 16X2 vista desde tu parte adversa. Fuente: El autor.

En la figura 62, se puede observar el sistema completo aun abierto dentro de su caja metálica

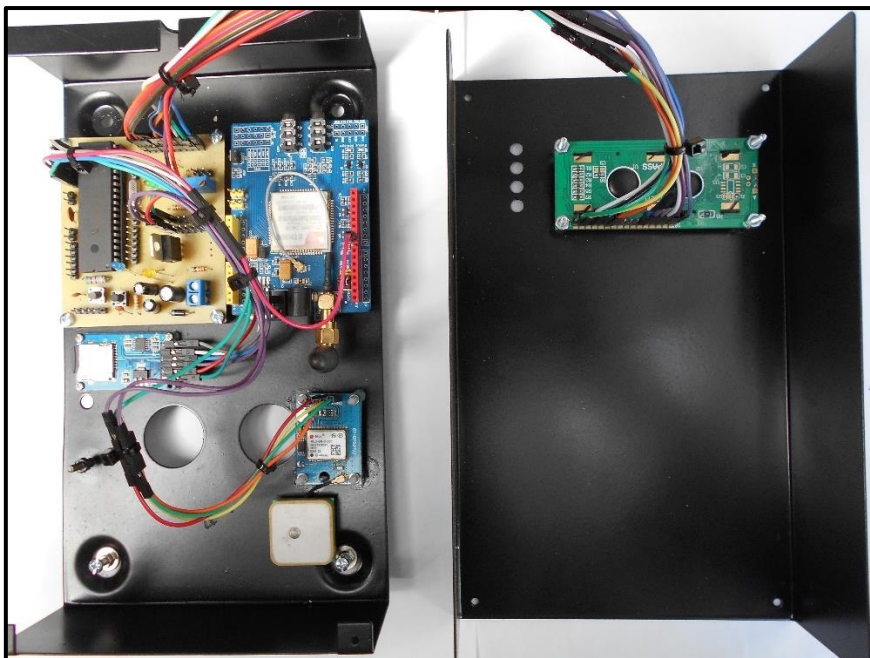


Figura 62. Sistema para la medición y registro de velocidad completo dentro de su caja metálica. Fuente: El autor.

Finalmente se muestra en la figura 63 el sistema completo visto desde su parte frontal.

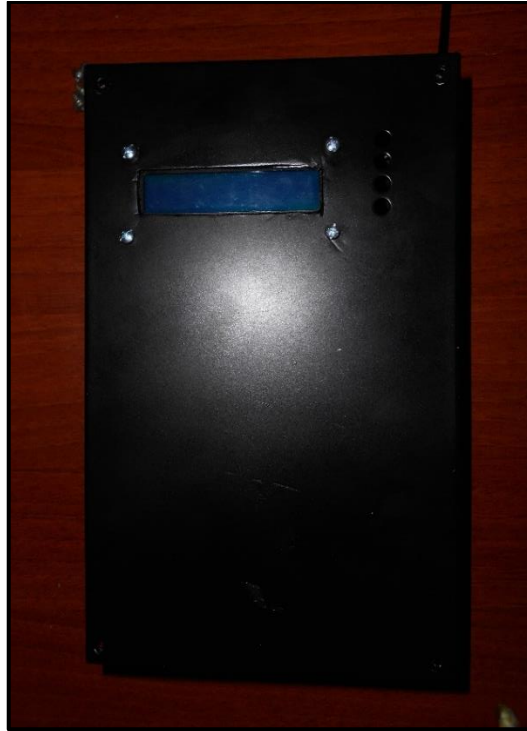


Figura 63 Sistema para medición y registro de velocidad mediante GPS visto desde su parte frontal.
Fuente: El autor.

Para el análisis de los datos obtenidos por el sistema antes mencionado, fue necesario la elaboración de un receptor para GPS, ya que los datos son enviados y por tanto recibidos en una PC donde se almacena esta información en forma de texto.

El elemento encargado de enviar los comandos AT al módulo SIM 800L (para receptor) es un Arduino Mega, el cual ejecuta un código que envía los comandos AT para visualizar los mensajes de texto en la pantalla de la PC y mediante un código desarrollado en Labview 2019, se almacena esta información en forma de texto. Esta descarga puede llevar alrededor de 10 minutos, que es el tiempo en el que llegan los 58 mensajes de texto al módulo SIM receptor. En la figura 64 se muestra el circuito receptor.

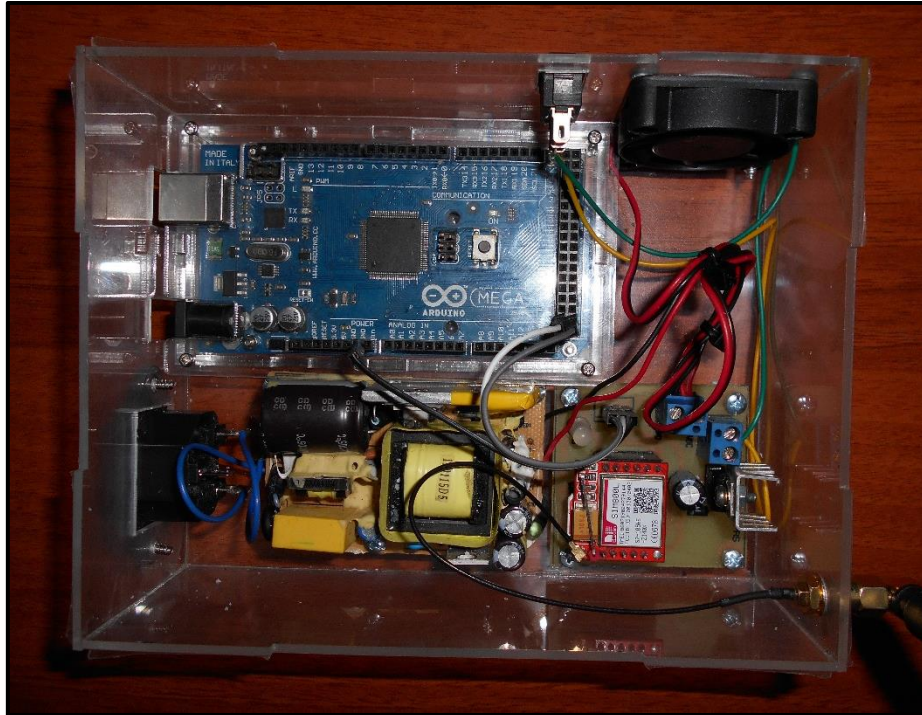


Figura 64. Conjunto de sub circuitos que conforman el receptor para visualizar los datos del GPS en la PC. Fuente. El autor.

Mediante el puerto USB del Arduino Mega podemos conectar el receptor a la PC, para luego visualizar los datos que posteriormente serán almacenados en forma de texto.

6.3 Resultados obtenidas en un vehículo.

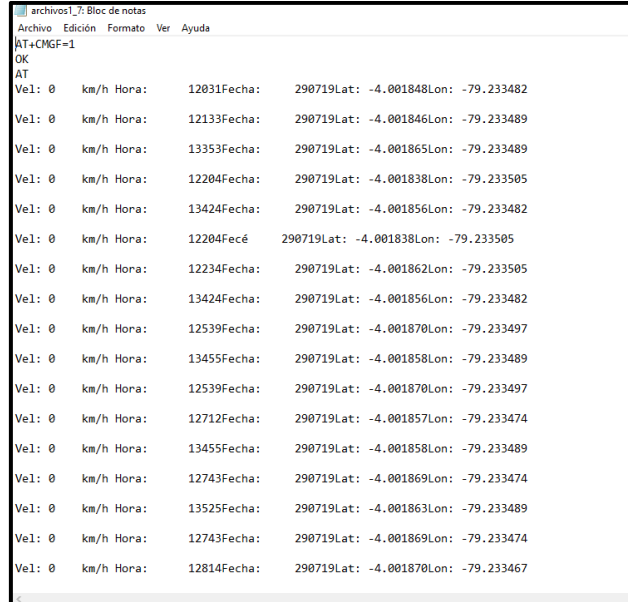
Luego de la elaboración de los respectivos circuitos, se procedió a la instalación del Circuito para la Medición y Registro de Velocidad mediante GPS donde se realizaron las respectivas pruebas. Mediante estas pruebas se validó el óptimo funcionamiento del circuito, la información aquí recopilada se detalla en las siguientes secciones.

6.4 Visualización de los datos obtenidos a través de la PC

Una vez que los datos se envían desde el Circuito para la Medición y Registro de velocidad, se puede conectar el circuito receptor mediante el puerto serial al USB de la PC y ejecutar el VI creado para este proyecto.

Como se mencionó en el apartado 5.8 (código para visualización de los datos obtenidos) los datos aquí obtenidos se guardan en un archivo de texto.

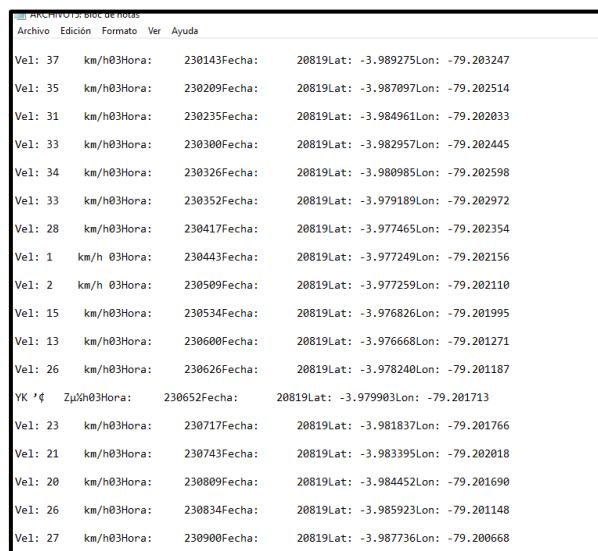
En la figura 65 se puede ver los datos obtenidos por un vehículo antes de empezar su desplazamiento. Como se puede ver se obtienen los datos de fecha, hora, velocidad y posición y se mantiene un registro de los desplazamientos realizados.



Vel	km/h	Hora	Fecha	290719Lat	-4.001848Lon	-79.233482
Vel: 0	km/h	Hora:	12031Fecha:	290719Lat:	-4.001848Lon:	-79.233482
Vel: 0	km/h	Hora:	12133Fecha:	290719Lat:	-4.001846Lon:	-79.233489
Vel: 0	km/h	Hora:	13353Fecha:	290719Lat:	-4.001865Lon:	-79.233489
Vel: 0	km/h	Hora:	12204Fecha:	290719Lat:	-4.001838Lon:	-79.233505
Vel: 0	km/h	Hora:	13424Fecha:	290719Lat:	-4.001856Lon:	-79.233482
Vel: 0	km/h	Hora:	12204Fecé	290719Lat:	-4.001838Lon:	-79.233505
Vel: 0	km/h	Hora:	12234Fecha:	290719Lat:	-4.001862Lon:	-79.233505
Vel: 0	km/h	Hora:	13424Fecha:	290719Lat:	-4.001856Lon:	-79.233482
Vel: 0	km/h	Hora:	12539Fecha:	290719Lat:	-4.001870Lon:	-79.233497
Vel: 0	km/h	Hora:	13455Fecha:	290719Lat:	-4.001858Lon:	-79.233489
Vel: 0	km/h	Hora:	12539Fecha:	290719Lat:	-4.001870Lon:	-79.233497
Vel: 0	km/h	Hora:	12712Fecha:	290719Lat:	-4.001857Lon:	-79.233474
Vel: 0	km/h	Hora:	13455Fecha:	290719Lat:	-4.001858Lon:	-79.233489
Vel: 0	km/h	Hora:	12743Fecha:	290719Lat:	-4.001869Lon:	-79.233474
Vel: 0	km/h	Hora:	13525Fecha:	290719Lat:	-4.001863Lon:	-79.233489
Vel: 0	km/h	Hora:	12743Fecha:	290719Lat:	-4.001869Lon:	-79.233474
Vel: 0	km/h	Hora:	12814Fecha:	290719Lat:	-4.001870Lon:	-79.233467

Figura 65. Datos almacenados en forma de texto obtenidos desde el GPS con un vehículo estático. Fuente: El autor.

En la figura 66 se puede visualizar los resultados obtenidos por un vehículo en movimiento. Estos datos se obtuvieron alrededor de las 22:00 horas hasta las 23:00. Se puede evidenciar que las coordenadas enviada corresponden al sitio en el que se llevó a cabo las pruebas.



Vel	km/h	Hora	Fecha	20819Lat	-3.989275Lon	-79.203247
Vel: 37	km/h	03Hora:	230143Fecha:	20819Lat:	-3.989275Lon:	-79.203247
Vel: 35	km/h	03Hora:	230209Fecha:	20819Lat:	-3.987097Lon:	-79.202514
Vel: 31	km/h	03Hora:	230235Fecha:	20819Lat:	-3.984961Lon:	-79.202033
Vel: 33	km/h	03Hora:	230300Fecha:	20819Lat:	-3.982957Lon:	-79.202445
Vel: 34	km/h	03Hora:	230326Fecha:	20819Lat:	-3.980985Lon:	-79.202598
Vel: 33	km/h	03Hora:	230352Fecha:	20819Lat:	-3.979189Lon:	-79.202972
Vel: 28	km/h	03Hora:	230417Fecha:	20819Lat:	-3.977465Lon:	-79.202354
Vel: 1	km/h	03Hora:	230443Fecha:	20819Lat:	-3.977249Lon:	-79.202156
Vel: 2	km/h	03Hora:	230509Fecha:	20819Lat:	-3.977259Lon:	-79.202110
Vel: 15	km/h	03Hora:	230534Fecha:	20819Lat:	-3.976826Lon:	-79.201995
Vel: 13	km/h	03Hora:	230600Fecha:	20819Lat:	-3.976668Lon:	-79.201271
Vel: 26	km/h	03Hora:	230626Fecha:	20819Lat:	-3.978240Lon:	-79.201187
YK '4	Zu	h03Hora:	230652Fecha:	20819Lat:	-3.979903Lon:	-79.201713
Vel: 23	km/h	03Hora:	230717Fecha:	20819Lat:	-3.981837Lon:	-79.201766
Vel: 21	km/h	03Hora:	230743Fecha:	20819Lat:	-3.983395Lon:	-79.202018
Vel: 20	km/h	03Hora:	230809Fecha:	20819Lat:	-3.984452Lon:	-79.201690
Vel: 26	km/h	03Hora:	230834Fecha:	20819Lat:	-3.985923Lon:	-79.201148
Vel: 27	km/h	03Hora:	230900Fecha:	20819Lat:	-3.987736Lon:	-79.200668

Figura 66. Datos obtenidos por un vehículo en movimiento. Fuente: El autor.

En la figura 67 se muestran los datos que son almacenados en la micro SD Card, inicialmente, con un vehículo estático, y luego con un vehículo en movimiento.

TEST.TXT: Bloc de notas									
Archivo	Edición	Formato	Ver	Ayuda					
Vel: 0	km/h	Hora:	190000	Fecha:	-10000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	140000	Fecha:	-10000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	90000	Fecha:	-10000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	40000	Fecha:	-10000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	230000	Fecha:	-20000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	180000	Fecha:	-20000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	130000	Fecha:	-20000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	80000	Fecha:	-20000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	30000	Fecha:	-20000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	220000	Fecha:	-30000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	170000	Fecha:	-30000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	120000	Fecha:	-30000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	70000	Fecha:	-30000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	20000	Fecha:	-30000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	210000	Fecha:	-40000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	160000	Fecha:	-40000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	110000	Fecha:	-40000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	60000	Fecha:	-40000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	10000	Fecha:	-40000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	200000	Fecha:	-50000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	150000	Fecha:	-50000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	100000	Fecha:	-50000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	50000	Fecha:	-50000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	240000	Fecha:	-60000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	430000	Fecha:	-70000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	620000	Fecha:	-80000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	810000	Fecha:	-90000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	1000000	Fecha:	-100000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	1190000	Fecha:	-110000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	1380000	Fecha:	-120000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	1570000	Fecha:	-130000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	1760000	Fecha:	-140000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	1950000	Fecha:	-150000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	2140000	Fecha:	-160000	Lat:	0.000000	Lon:	0.000000
Vel: 0	km/h	Hora:	2330000	Fecha:	-170000	Lat:	0.000000	Lon:	0.000000

Figura 67 Datos almacenados en la micro SD Card con vehículo estático. Fuente: El autor.

Mientras no se estable la comunicación, la micro SD Card guarda datos en cero, durante los primeros minutos, posteriormente se comienzan almacenar los datos de ubicación, esto se muestra en la figura 68.

TEST.TXT: Bloc de notas								
Archivo	Edición	Formato	Ver	Ayuda				
Vel:	Hora:	183629	Fecha:	40819	Lat:	-4.003710	Lon:	-79.19697504
Vel:	Hora:	183629	Fecha:	40819	Lat:	-4.003710	Lon:	-79.19697504
Vel:	Hora:	183648	Fecha:	40819	Lat:	-4.003723	Lon:	-79.19696004
Vel:	Hora:	183700	Fecha:	40819	Lat:	-4.003731	Lon:	-79.19696004
Vel:	Hora:	183713	Fecha:	40819	Lat:	-4.003743	Lon:	-79.19697504
Vel:	Hora:	183725	Fecha:	40819	Lat:	-4.003748	Lon:	-79.19698304
Vel:	Hora:	183738	Fecha:	40819	Lat:	-4.003754	Lon:	-79.19698304
Vel:	Hora:	183750	Fecha:	40819	Lat:	-4.003756	Lon:	-79.19698304
Vel:	Hora:	183803	Fecha:	40819	Lat:	-4.003763	Lon:	-79.19698304
Vel:	Hora:	183815	Fecha:	40819	Lat:	-4.003774	Lon:	-79.19697504
Vel:	Hora:	183828	Fecha:	40819	Lat:	-4.003782	Lon:	-79.19697504
Vel:	Hora:	183841	Fecha:	40819	Lat:	-4.003781	Lon:	-79.19696804
Vel:	Hora:	183853	Fecha:	40819	Lat:	-4.003779	Lon:	-79.19696804
Vel:	Hora:	183906	Fecha:	40819	Lat:	-4.003787	Lon:	-79.19698304
Vel:	Hora:	183918	Fecha:	40819	Lat:	-4.003796	Lon:	-79.19697504
Vel:	Hora:	183931	Fecha:	40819	Lat:	-4.003799	Lon:	-79.19696004
Vel:	Hora:	183943	Fecha:	40819	Lat:	-4.003804	Lon:	-79.19695204
Vel:	Hora:	183956	Fecha:	40819	Lat:	-4.003806	Lon:	-79.19693704
Vel:	Hora:	184008	Fecha:	40819	Lat:	-4.003805	Lon:	-79.19694504
Vel:	Hora:	184021	Fecha:	40819	Lat:	-4.003797	Lon:	-79.19694504
Vel:	Hora:	184033	Fecha:	40819	Lat:	-4.003787	Lon:	-79.19696004
Vel:	Hora:	184046	Fecha:	40819	Lat:	-4.003786	Lon:	-79.19696004
Vel:	Hora:	184058	Fecha:	40819	Lat:	-4.003781	Lon:	-79.19696804
Vel:	Hora:	184111	Fecha:	40819	Lat:	-4.003771	Lon:	-79.19698304
Vel:	Hora:	184124	Fecha:	40819	Lat:	-4.003765	Lon:	-79.19699004
Vel:	Hora:	184136	Fecha:	40819	Lat:	-4.003770	Lon:	-79.19697504
Vel:	Hora:	184149	Fecha:	40819	Lat:	-4.003776	Lon:	-79.19696004
Vel:	Hora:	184201	Fecha:	40819	Lat:	-4.003779	Lon:	-79.19695204
Vel:	Hora:	184214	Fecha:	40819	Lat:	-4.003776	Lon:	-79.19696804
Vel:	Hora:	184226	Fecha:	40819	Lat:	-4.003776	Lon:	-79.19697504
Vel:	Hora:	184239	Fecha:	40819	Lat:	-4.003777	Lon:	-79.19697504
Vel:	Hora:	184251	Fecha:	40819	Lat:	-4.003777	Lon:	-79.19697504
Vel:	Hora:	184304	Fecha:	40819	Lat:	-4.003774	Lon:	-79.19698304
Vel:	Hora:	184316	Fecha:	40819	Lat:	-4.003785	Lon:	-79.19697504

Figura 68 Datos de ubicación con vehículo estático. Fuente: El autor.

6.5 Visualización mediante plataforma de posicionamiento

Como se mencionó en la sección 5.8.3 (desarrollo de plataforma de posicionamiento), las coordenadas de posicionamiento pueden ser visualizadas a través de una interfaz, la cual

opera mediante la plataforma de Google Maps. Esta herramienta web permite visualizar los datos obtenidos por el módulo u-blox 6 en conjunto con Labview, de esta manera se puede obtener una representación más exacta y grafica de los datos recibidos.

Tomando en cuenta que siempre existirán dos estados, en el que se envía una coordenada (estado uno) y en la que no se envía una coordenada (estado cero). Durante el estado en el que no se envía una coordenada, se mostrara el mapa del país, ya que el bloque de Url de Labview solo mostrara el enlace principal.

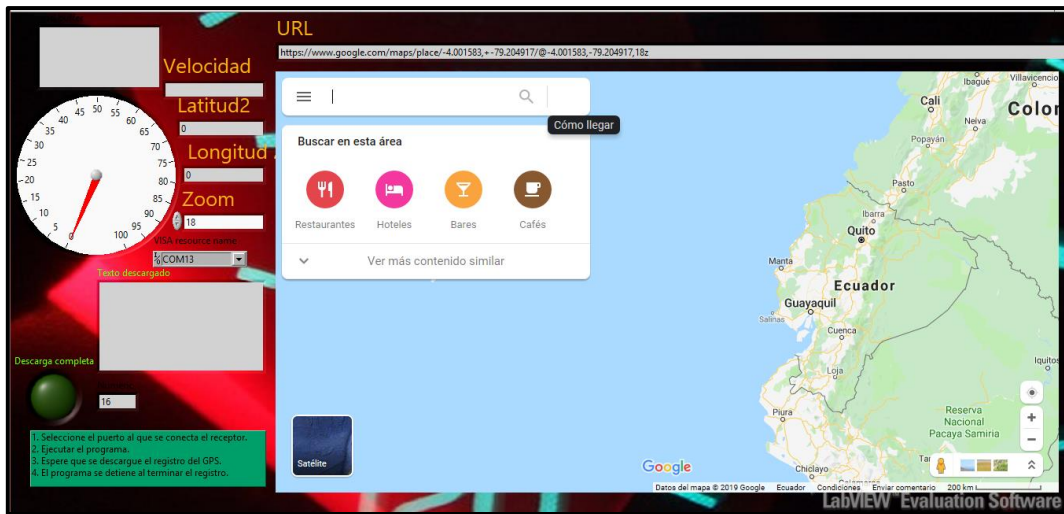


Figura 69. vista principal de la plataforma, en el estado 0. Fuente: El autor.

En la figura 70 se muestra las coordenadas dibujadas en el estado 1, cuando se tiene datos que pueden ser graficados. Mediante este método se va graficando los dos estados luego de obtener los datos en el SIM800L.

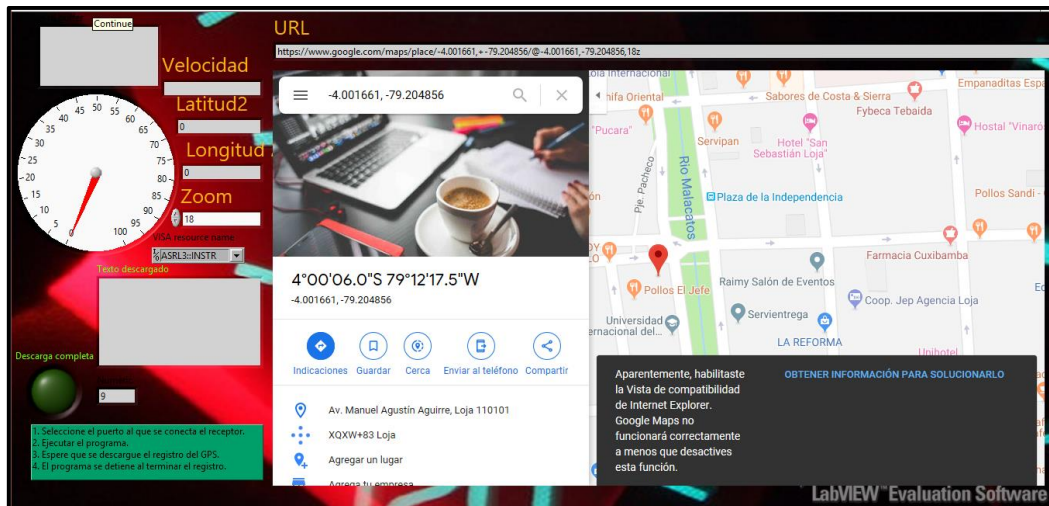


Figura 70. Punto graficado obtenido Av. Manuel Agustín Aguirre. Fuente: El autor.

En la figura 71 se muestra las coordenadas de un punto obtenido en el estado uno, luego comparamos este punto con una aplicación desde el teléfono móvil.

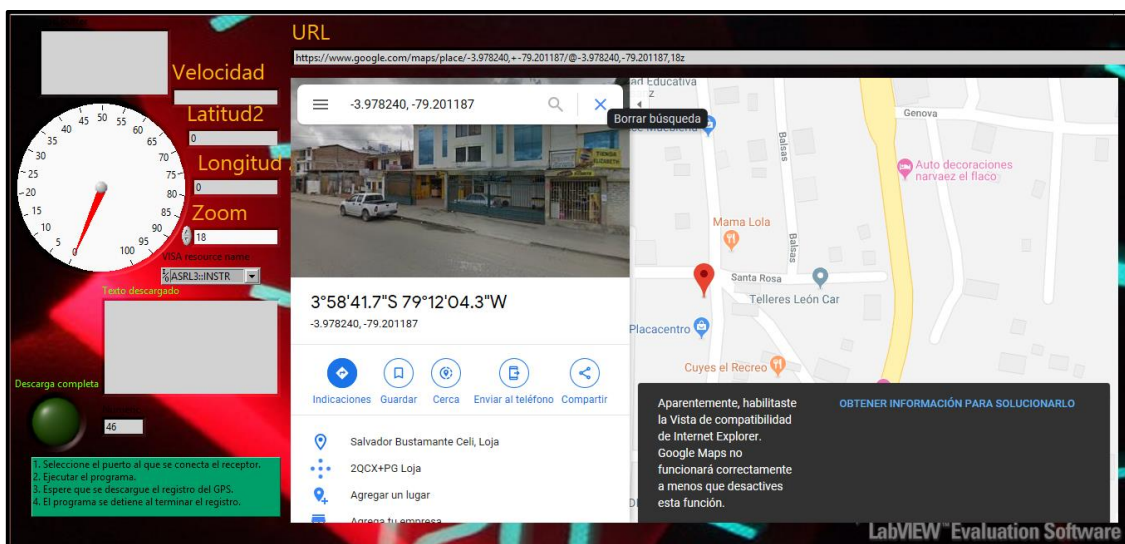


Figura 71 Coordenadas obtenidas desde la plataforma. Fuente: El autor.

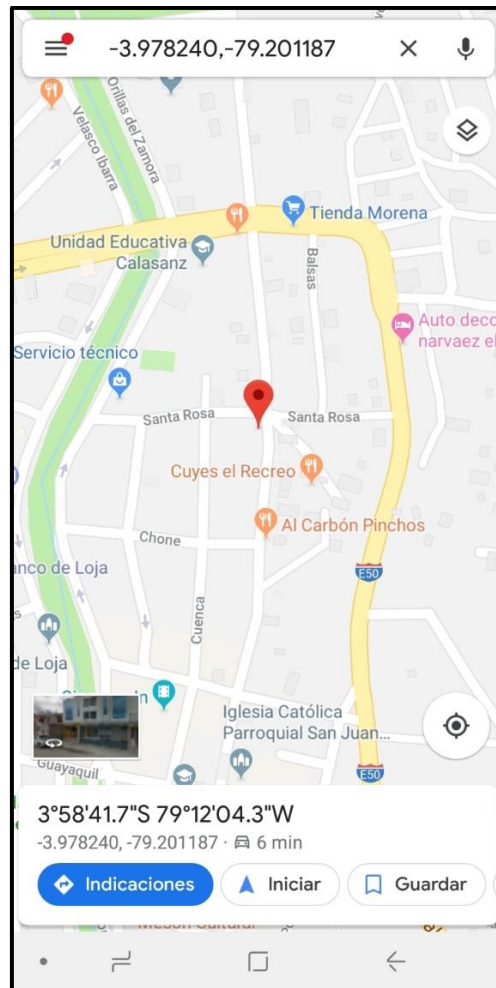


Figura 72. Imagen obtenida desde la aplicación de Google Maps. Fuente: El autor.

Por lo tanto, podemos concluir que los dos estados del sistema conforman la totalidad para su funcionamiento, en la figura 73 y 74, podemos ver el punto que define las coordenadas en el mapa de Google Maps y los datos de velocidad, fecha y hora, en la que se tomó la muestra de posición.

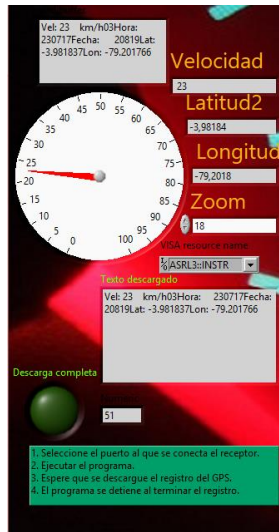


Figura 73. Datos de posición, fecha, hora y velocidad obtenidos desde el SIM800L. Fuente: El autor.

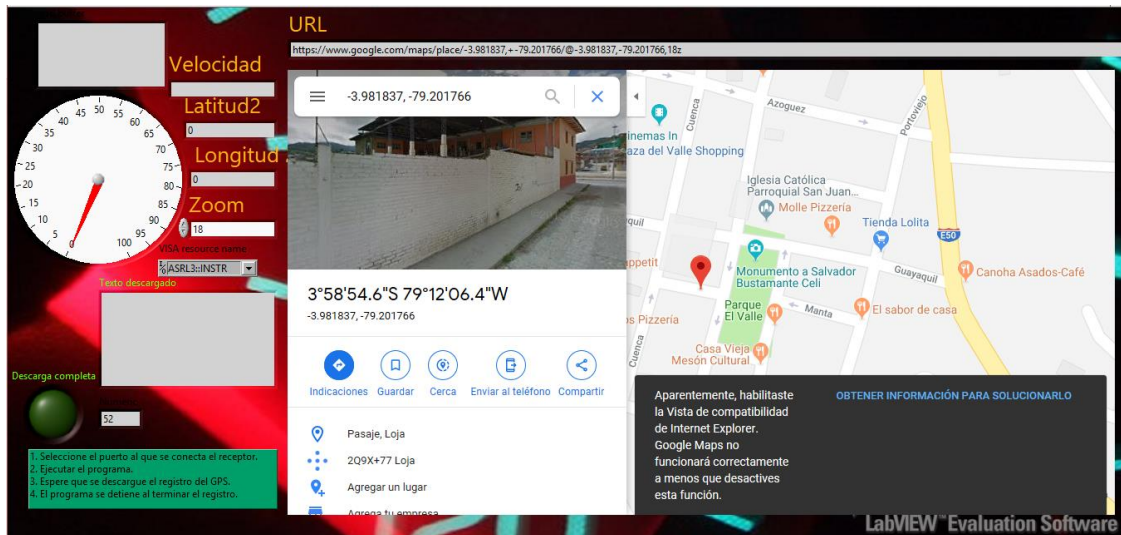


Figura 74. Posición graficada en Google Maps. Fuente: El autor.

7. DISCUSIÓN

En los objetivos de este proyecto constan la implementación de un circuito que posea las características de medir velocidad, registrarla y enviarla. Dentro de este campo; esto se llevó a cabo mediante el uso del módulo GPS U-blox 6 que recibe la señal proveniente de los distintos satélites a los que se conecta. La señal que recibe posee el formato NMEA 2000 por lo que hay que procesar esta información mediante el PIC184620 para que tenga una forma leíble al usuario, luego la información es almacenada en una microSD Card de 4 Gb y enviada mediante el SIM900L.

El segundo objetivo planteaba la instalación del sistema desarrollado en un vehículo para validar su funcionamiento, este objetivo se desarrolló en dos horarios, durante la tarde y durante la noche, las coordenadas obtenidas coincidieron con las coordenadas conseguidas mediante una aplicación que se usó como referencia para comprobar su óptimo funcionamiento.

Para el tercer objetivo se realizó una plataforma de posicionamiento que combina las propiedades de Labview y Google maps, para poder graficar los puntos de manera que se pueda tener una mejor idea del sitio en el que el usuario se encontraba a cierta hora, cabe destacar que no fue necesario el uso de la librería Gmaps de Labview, ya que para el uso de esta librería se requiere una clave API proporcionada por Google Maps, la cual tiene una limitante de tiempo de tres meses de funcionamiento. Mediante el uso de las propiedades de URLs de Labview se puede concatenar directorios hasta lograr una dirección web para graficar el punto de posición del vehículo, así como visualizar mediante el panel de control la velocidad, fecha y hora del vehículo.

El cuarto objetivo que se planteó para este proyecto indicaba proponer ante los organismos pertinentes la homologación del dispositivo para que pueda ser usado por los consumidores como una herramienta de defensa ante posibles multas erróneas. El organismo regulador de esta instancia es la Agencia Nacional de Tránsito. El dispositivo fue presentado con la documentación requerida por el departamento tecnológico de la agencia nacional de tránsito para su envío a la ciudad de Quito y posteriormente entrar a un proceso de homologación. La documentación requerida consta de básicamente de hojas de datos de los diferentes módulos, estructura general del sistema y fotos que corroboren el funcionamiento del sistema.

8. CONCLUSIONES

- ✓ Las limitaciones del sistema de fotomultas instalado actualmente en la ciudad de Loja, permite el desarrollo e investigación de nuevas fuentes tecnológicas mediante la cual se pueden verificar el exceso de velocidad, como en este caso el uso de la tecnología del GPS.
- ✓ El margen de error bajo del GPS (cerca de 10 a 2 metros) permite tener un resultado óptimo para que esta tecnología sea útil como complemento del sistema de fotomultas en la ciudad de Loja
- ✓ Al momento de desarrollar un circuito basado en GPS, es necesario tener en cuenta los datasheets que presenta el fabricante ya que mediante estos se puede obtener información de pérdidas, tiempo de conexión y ganancia en la antena, lo que es indispensable para el normal desarrollo del sistema.
- ✓ Es importante el análisis del uso de puertos SPI mediante el cual se pueden enviar datos a otro PIC, dejando libres puertos seriales que pueden ser usados para otros periféricos.
- ✓ Los resultados obtenidos en la placa de pruebas sirvieron para dar un mejor enfoque en los resultados alcanzados en la placa final, ya que en la primera se despejaron posibles dudas que se iban presentando en el desarrollo de este proyecto
- ✓ La documentación del dispositivo queda a órdenes del organismo regulador que en este caso es la Agencia Nacional de Transito, para su posterior análisis para entrar al proceso de homologación y ser usado de manera general por los usuarios.
- ✓ La visualización de los datos de posición en una plataforma permite dar una mejor idea del sitio en el que se ubica el vehículo mostrando también información de hora, velocidad y fecha.
- ✓ Las pruebas efectuadas arrojan como resultado un error que puede ser desapercibido al momento de incorporar el sistema a un funcionamiento complementario al sistema de fotomultas.

9. RECOMENDACIONES

- ✓ Se recomienda el uso de nuevas tecnologías para la obtención de información de exceso de velocidad para los usuarios de vehículos infractores, ya que el sistema actual solo maneja sensores piezoeléctricos que necesitan constante mantenimiento por su desgaste. En algunas ciudades como Cuenca el sistema principalmente está basado en radares que necesitan menor mantenimiento.
- ✓ El uso del GPS como complemento del sistema de fotomultas en Loja se puede ampliar de tal forma que opere de manera jerárquica al sistema de fotomultas si se lograra un sistema integrado para enlazar la información obtenida por el sistema.
- ✓ La inversión a gran escala en proyectos de este tipo puede mejorar notablemente el desempeño del sistema, ya que se podrían obtener módulos con mayor velocidad de conexión y así mismo reduciría el margen de error.
- ✓ Se recomienda tomar en consideración los datos de corriente de los diferentes módulos, ya que durante los primeros segundos en los que se establece la comunicación, el nivel de corriente consumido es mayor por lo que se debe tener buen suministro de corriente.
- ✓ Se recomienda trabajar con Google maps sin asociarse a un cuenta, ya q esto permitirá visualizar los datos desde cualquier otra plataforma q maneje formato CSV.

10.BIBLIOGRAFÍA

- [1] Agencia Nacional de transito , «tabulados del reporte nacional de siniestros de transito, enero 2019,» Agencia Nacional de Tránsito - ANT, Quito, 2019.
- [2] Y. Diaz, «MÁS DE QUINIENTOS SANCIONADOS POR EXCEDER VELOCIDAD,» 14 02 2019. [En línea]. Available: <http://www.loja.gob.ec/noticia/2019-02/mas-de-quinientos-sancionados-por-exceder-velocidad>. [Último acceso: 15 03 2019].
- [3] A. Vallejo, «actualidad sistema de fотomultas,» 13 enero 2019. [En línea]. Available: <https://www.elcomercio.com/actualidad/sistema-fotomultas-inicio-activacion-quito.html>.
- [4] Agencia Nacional de Transito, «direccion de regulacion homologacion de equipos y dispositivos de control,» 03 09 2018. [En línea]. Available: <https://www.ant.gob.ec/index.php/descargable/file/5749-listado-de-dispositivos-homologados-al-03-09-2018>.
- [5] M. C. O. Sandoval, «repositorio universidad nacional de Loja,» 16 8 2017. [En línea]. Available: <http://dspace.unl.edu.ec/jspui/bitstream/123456789/17913/1/Mar%C3%ADa%20Cecilia%20Ontaneda%20Sandoval.pdf>. [Último acceso: 29 4 2019].
- [6] Speed Cameras , «tipos de camaras de velocidad,» 08 02 2019. [En línea]. Available: <https://www.speedcamerasuk.com/speedcurb.htm>.
- [7] Safety enforcement seguridad vial , «camaras fijas y moviles,» 28 septiembre 2018. [En línea]. Available: <http://www.ses.com.ec/camaras-fijas/>.
- [8] L. Castillo, «El comercio,» 03 06 2017. [En línea]. Available: <https://www.elcomercio.com/audios/polemica-loja-limites-velocidad-fotomultas.html>.
- [9] R. R. SpeedCurb, «antiradares,» 5 10 2007. [En línea]. Available: <http://portalvasco.com/blog/2007/10/radar-redspeed-speedcurb/>. [Último acceso: 11 6 2019].
- [10] J. T. A. J. M. M. Daniel Sastre Garcia, «Cuaderno tecnologico de la PTC,» de *Sistema de adquisicion de informacion de trafico: estado actual y futuro*, Madrid, Tecnológica Española, 2011, p. 16.
- [11] Safety Enforcement, «Reglamento a la ley de transporte,» s/f s/f s/f. [En línea]. Available: <http://www.ses.com.ec/reglamento-a-la-ley-de-transporte-terrestre/>. [Último acceso: 11 6 19].

- [12] Agencia Nacional de tránsito, «fotoradores de última tecnología para el control de velocidad,» s/f s/f s/f. [En línea]. Available: <https://www.ant.gob.ec/index.php/component/content/article/49-boletines/189-agencia-nacional-de-transito-participa-en-la-semana-#.XQQBUohKjIU>. [Último acceso: 14 6 2019].
- [13] RTC Manufacturing, Inc, «Specifications Exhibits Group E,» s/f s/f s/f. [En línea]. Available: http://www.ledmovie.com/docs/speed_sign_manual_EN.pdf. [Último acceso: 14 6 2019].
- [14] la hora, «Fotomultas: conozca las maneras de notificar,» 15 enero 2019. [En línea]. Available: <https://www.pressreader.com/ecuador/la-hora-loja/20190115>.
- [15] J. D. J. Q. Erika Cajamarca, «diseño e implementación de un sistema de medición de velocidad, registro y monitoreo de movimiento del día para un vehículo diesel,» Bogotá, 2006, p. 31.
- [16] V. Pospelova, «concurso 2014,» 16 febrero 2014. [En línea]. Available: <http://www3.uah.es/libretics/concurso2014/files2014/Trabajos/Posicionamiento.pdf>.
- [17] G. Threepwood, «youbioit,» 27 10 2013. [En línea]. Available: https://youbioit.com/es/article/15626/que-es-la-trilateracion-satelital&size=_original.
- [18] J. D. T. Alberto Eduardo Riva, «Estandar NMEA-0183,» de *Procesamiento de sentencias NMEA-0183 para el análisis de la geometría satelital utilizando receptores GPS de bajo costo*, Buenos Aires-Argentina, s/e, 2016, p. 1027.
- [19] E. G. D. T. Juan Pablo Astudillo, sistema de localización y monitoreo y control vehicular basada en los protocolos GPS/GSM/GPRS, Cuenca, 2012.
- [20] A. Ponce, «¿Que es GPRS?,» 2 marzo 2019. [En línea]. Available: <https://tecnosinergia.zendesk.com/hc/es/articles/236174548--Qu%C3%A9-es-GPRS->.
- [21] e. d. e. d. u. i. d. Valencia, «Universidad Internacional de Valencia,» 21 03 2018. [En línea]. Available: <https://www.universidadviu.com/sistemas-gprs-concepto-general-usos-principales/>.
- [22] D. P. Valdés, «Maestros del Web,» 26 10 2007. [En línea]. Available: <http://www.maestrosdelweb.com/que-son-las-bases-de-datos/>.
- [23] A. S. Uzabal, «Autonomía Digital y tecnológica,» 21 06 2018. [En línea]. Available: <https://voragine.net/usando-un-archivo-de-texto-plano-como-base-de-datos-alternativa-ligera-a-sql>. [Último acceso: 04 08 2019].

- [24] Arduino.cc, Arduino Uno, Italia.
- [25] Microchip Technology Inc, Datasheet Enhanced Flash Microcontrollers with 10-Bit A/D and nanoWatt Technology, Arizona, Estados Unidos., 2008.
- [26] Copyright © Parallax Inc, Parallax's PAM-7Q GPS, Roseville, California, 2014.
- [27] u-blox, «Neo-6,» 2011. [En línea]. Available: <https://www.u-blox.com>.
- [28] SIMCom , «SIM900 Reference Desing Guide,» 14 08 2010. [En línea]. Available:
https://simcom.ee/documents/SIM900/AN_SIM900%20Reference%20Design%20Guide_V1.02.pdf. [Último acceso: 10 02 2019].
- [29] SIMCom, «SIM800 Series _Eembedded V1.01,» 11 07 2013. [En línea]. Available:
https://simcom.ee/documents/SIM800x/SIM800%20Series%20Embedded%20AT_Compiling%20Environment_V1.01.pdf. [Último acceso: 15 01 2019].
- [30] Vishnu Mia, «sitio web del proyecto,» Módulo de tarjeta Micro SD Catalex de interfaz con Arduino, 09 03 2017. [En línea]. Available:
<https://www.vishnumaiea.in/projects/hardware/interfacing-catalex-micro-sd-card-module>. [Último acceso: 15 03 2019].
- [31] electronilab, «Diagramas,» 2018. [En línea]. Available:
<https://electronilab.co/tienda/modulo-gps-ublox-neo-6m-v2-con-memoria-eeeprom/>.

11.ANEXOS

ANEXO 1: CÓDIGO EN MIKROC PARA PIC 18F4620

```
// =====  
//  
// LIBRERIA LCD  
  
sbit LCD_RS at RD4_bit;  
sbit LCD_EN at RD5_bit;  
sbit LCD_D4 at RD0_bit;  
sbit LCD_D5 at RD1_bit;  
sbit LCD_D6 at RD2_bit;  
sbit LCD_D7 at RD3_bit;  
  
sbit LCD_RS_Direction at TRISD4_bit;  
sbit LCD_EN_Direction at TRISD5_bit;  
sbit LCD_D4_Direction at TRISD0_bit;  
sbit LCD_D5_Direction at TRISD1_bit;  
sbit LCD_D6_Direction at TRISD2_bit;  
sbit LCD_D7_Direction at TRISD3_bit;  
  
// FIN LIBRERIA LCD  
// =====  
  
typedef unsigned short uint8;  
typedef signed short int8;  
  
static const uint8  
FILE_READ = 0x01,  
FILE_WRITE = 0x02,  
FILE_APPEND = 0x04;  
  
// Mmc Library requirements  
sbit Mmc_Chip_Select at RC2_bit;  
sbit Mmc_Chip_Select_Direction at TRISC2_bit;  
  
// SPI initialization routines required by SD/Mmc Library  
void initSPI(void)  
{  
SPI1_Init_Advanced(_SPI_MASTER_OSC_DIV64, _SPI_DATA_SAMPLE_MIDDLE, _SPI_CLK_IDLE_LOW,  
_SPI_LOW_2_HIGH);  
}  
  
void initFastSPI(void)
```

```

{
SPI1_Init_Advanced(_SPI_MASTER_OSC_DIV4, _SPI_DATA_SAMPLE_MIDDLE, _SPI_CLK_IDLE_LOW,
_SPI_LOW_2_HIGH);
}

int8 err;

short fhandle[4]; // file handle variable should be signed (4 = max files can be open at
the same time)

// =====
//
//                               VARIABLES GLOBALES
//
char txt[768];
char texto_velo [15];
char texto_hora [15];
char texto_fecha[15];
char texto_latitud[15];
char texto_longitud[15];
char texto_auxiliar[15];
char texto_grados[15];
char latitudf[20];
char gradosf[20];

char *velo;
char *string;
unsigned short listo;
int i;

//
//                               FIN DE VARIABLES
//
// =====

// =====
//
//                               FUNCION PARA INTERRUPCION
//
void interrupt()
{
    if(PIR1.F0 == 1) // Ha habido desbordamiento en Timer1?
    {
        T1CON.F0 = 0; // Desactivar Timer1
        listo = 1; // Datos listos
        i = 0; // Reiniciar interacciones de cadena
        PIR1.F0 = 0; // Limpiar bandera
    }
}

```

```

    } //end if TMR1IF

    if (PIR1.F5 == 1) // ¿Hubo una interrupción de la UART?
    { // Si
        txt[i++] = UART1_Read(); // lee y asigna al elemento siguiente de
        la cadena
        LATB5_bit = ~LATB5_bit; // Invierte el estado de RB0
        if(i == 768) i = 0; // ¿llegó a 768? vuelve a ser cero
        T1CON.F0 = 0; // Desactivar Timer1
        TMR1H = 0x0B; // Interrupcion cada 100 ms
        TMR1L = 0xDC; // Interrupcion cada 100 ms
        T1CON.F0 = 1; // Activa Timer1
        PIR1.F5 = 0; // Limpia la bandera RCIF
    } //end if RCIF
} //end interrupt
//
// =====

void mensaje_inicio(void)
{
    Lcd_Out(1,3,"PIC_18F4620"); // Envia mensaje
    Lcd_Out(2,2,"Receptor GPS"); // Envia mensaje
    Delay_ms(1000); // Espera 1 segundo
    Lcd_Cmd(_LCD_CLEAR); // Limpia LCD
    Lcd_Out(1,3,"Esperando"); // Envia mensaje
    Lcd_Out(2,4,"conexion..."); // Envia mensaje
    Delay_ms(1000); // Espera 1 segundo
}

void configuracion_puertos(void)
{
    ADCON1 = 0x0F; // (0xFF) Desactiva las entradas
    analogicas y las configura como salidas digitales.
    TRISB6_bit = 1; // ENTRADA PULSADOR
    TRISB4_bit = 0; // LED 0 - sd fail
    TRISB5_bit = 0; // LED 1 - comunicacion ok
    TRISB7_bit = 0; // LED 2 - sd ok
    //LATB7_bit = 0; // RB7 inicia en low 0
    //LATB5_bit = 0; // RB5 inicia en low 0
    //LATB4_bit = 0; // RB4 inicia en low 0

```



```

}

void configuracion_interupcion(void)
{
    listo      = 0;                // limpia listo o lo vuelve cero "0"
    T1CON.F5 = 1;                // TCKPS1 = 1;   Timer1 con prescaler 1:8
    T1CON.F4 = 1;                // TCKPS0 = 1   Timer1 con prescaler 1:8
    PIE1.F0 = 1;                // TMR1IE = 1;  Habilita interrupcion de
Timer1
    TMR1H     = 0x0B;            // Interrupcion cada 100 ms
    TMR1L     = 0xDC;            // Interrupcion cada 100 ms
    PIR1.F0 = 0;                // TMR1IF = 0;   Limpia flag de Timer1
    UART1_Init(9600);           // inicializa UART1 em 9600 baud rate
    Delay_ms(100);              // Pausa de 100 ms
    PIE1.F5 = 1;                // RCIE = 1;    Habilita recepcion de
UART:
    INTCON.F7 = 1;              // GIE = 1 habilita interrupcion global
    INTCON.F6 = 1;              // PEIE = 1 habilita interrupcion por
periféricos
    T1CON.F0 = 1;                // Activa Timer1
}

```

```

void inicio_lcd(void)
{
    Lcd_Init();
    Delay_ms(50);                // inicializa LCD
    Lcd_Cmd(_LCD_CURSOR_OFF);   // Desactiva o apaga cursor
    Lcd_Cmd(_LCD_CLEAR);        // Limpia LCD
}

```

```

// =====
//          Función para buscar y filtrar un campo en la trama.
//
void BuscarCampo( char *Fuente, char *Destino, short campo )
{
    unsigned short j=0,k=0;
    //Bucle para buscar el campo.
    while( j<campo )if( Fuente[k++]=='.' )j++;
    j=0;
}

```

```

//Bucle para copiar el campo.
while( !(Fuente[k]==' ' || Fuente[k]=='*') )
{
    Destino[j++]=Fuente[k++];
    Destino[j]=0;
}
}
//
// =====

// =====
//
//          FUNCION VELOCIDAD          ////////////////
void Velocidad_kmh()
{
    unsigned short velocity;

    BuscarCampo(velo,texto_velo,7);        // Busca el campo de la velocidad en Km/h
    velocity = atof(texto_velo);           // Convierte de cadena a numero
    Delay_ms(20);                          // Espera 20 us
    ShortToStr(velocity,texto_velo);        // Convierte el valor de velocidad a cadena
    Lcd_Cmd(_LCD_CLEAR);                   // Limpia LCD
    Lcd_Out(1,1, "Vel:");                  // Envia mensaje
    Lcd_Out_CP(texto_velo);                 // Muestra la velocidad en LCD
    Lcd_Out_CP(" km/h");                   // Envia mensaje
    Delay_ms(2000);                        // Espera 2 s.
    if(velocity > 2)LATB7_bit = 1;          // Si la velocidad es mayor a 2 km/h prende
    el led,
    else{                                    // Caso contrario,
        LATB7_bit = 0;                      // Apaga el LED de RB6.
    }
} // end Velocidad_kmh
//
// =====

// =====
//
//          FUNCION HORA - FECHA
void Hora_Fecha()
{
    float h;

```

```

float date;

    BuscarCampo(string,texto_hora,1);           // Busca el campo de la hora UTC
    h = atof(texto_hora);                       // Convierte el valor de cadena a
numero                                         // numero
    BuscarCampo(string,texto_fecha,9);         // Busca el campo de la Fecha.
    date = atof(texto_fecha);                  // Convierte el valor de cadena a
numero                                         // numero

    if(h >= 060000.00 && h <= 235900.00)       // Si la hora UTC es mayor o igual de
06:00:00 y menor que 23:59:59,
    {                                           // entonces,
        h = h - 050000.00;                     // la hora local "Ecuador" será
05:00:00 menos que la hora UTC
    }
    else                                       // Caso contrario,
    {                                           // la hora local "Ecuador" se
        h = h + 190000.00;                     // la hora local "Ecuador" se
05:00:00 más que la hora UTC,
        date = date - 010000.00;               // y la fecha local será 1 día menos
que la fecha del GPS.
    }

    LongToStr(h,texto_hora);                   // Convierte en cadena el valor
numérico de la hora
    LongToStr(date,texto_fecha);               // Convierte en cadena el valor
numérico de la fecha

    Lcd_Cmd(_LCD_CLEAR);                       // Limpia o borra el contenido de la
LCD
    Lcd_Out(1,1,"Fecha: ");                    // Muestra la Fecha en el LCD
    Lcd_Chr(1,8,texto_fecha[5]);              //
    Lcd_Chr(1,9,texto_fecha[6]);              //
    Lcd_Chr_CP ('-');                          //
    Lcd_Chr(1,11,texto_fecha[7]);              //
    Lcd_Chr(1,12,texto_fecha[8]);              //
    Lcd_Chr_CP ('-');                          //
    Lcd_Chr(1,14,texto_fecha[9]);              //
    Lcd_Chr(1,15,texto_fecha[10]);             //

    Lcd_Out(2,1,"Hora: ");                     // Muestra la Hora en el LCD
    Lcd_Chr(2,7,texto_hora[5]);                //
    Lcd_Chr(2,8,texto_hora[6]);                //
    Lcd_Chr_CP (':');                          //

```

```

    Lcd_Chr(2,10,texto_hora[7]);           //
    Lcd_Chr(2,11,texto_hora[8]);         //
    Lcd_Chr_CP (':');                     //
    Lcd_Chr(2,13,texto_hora[9]);         //
    Lcd_Chr(2,14,texto_hora[10]);        //

    Delay_ms(2000);                       // Espera 2 segundos.

//-----
/*Lcd_Out(1,1,"UTC: ");
    Lcd_Chr_CP ( ' ');
    Lcd_Chr(1,6,string[7]);
    Lcd_Chr(1,7,string[8]);
    Lcd_Chr_CP (':');
    Lcd_Chr(1,9,string[9]);
    Lcd_Chr(1,10,string[10]);
    Lcd_Chr_CP (':');
    Lcd_Chr(1,12,string[11]);
    Lcd_Chr(1,13,string[12]);
    //Delay_ms(2000);

    Lcd_Out(2,1,"Date: ");
    Lcd_Chr(2,8,string[53]);
    Lcd_Chr(2,9,string[54]);
    Lcd_Chr_CP ('/');
    Lcd_Chr(2,11,string[55]);
    Lcd_Chr(2,12,string[56]);
    Lcd_Chr_CP ('/');
    Lcd_Chr(2,14,string[57]);
    Lcd_Chr(2,15,string[58]);*/

//-----
} // end hora
//
// =====

// =====
//
//                               FUNCION LATITUD - LONGITUD
void Latitud_Longitud()
{

```

```

float latitud;
float longitud;
float grados;
float minutos;
float grados2;
float minutos2;

//////////////////////////////////////          LATITUD          ////////////////////////////////////////
memset(gradosf ,0,15);                          // limpia los datos de grados...
memset(texto_latitud ,0,15);                      // limpia los datos de latitud...
BuscarCampo(string,texto_latitud,3);              // Busca el campo de la latitud
strncpy(gradosf,texto_latitud,2);                // captura los grados
grados2 = atof(gradosf);                          // FloatToStr(latitud, texto_latitud);
Delay_us(20);                                     // Pausa de 20 us
minutos2=atof(&texto_latitud[2])/60.0;            // apunta a los minutos
latitud=grados2+minutos2;                         // los pasa a grados
//FloatToStr(latitud,latitudf);                  // Convierte el valor de velocidad a cadena

BuscarCampo(string,texto_latitud,4);              // Busca el campo de la coordenada
N o S
if (texto_latitud[0]=='S') latitud=-latitud;      // Si apunta hacia el S entonces
sera negativa la latitud

sprintf(latitudf,"Lat: %f",latitud);              // ACONDICIONA PARA EL LCD
Lcd_Cmd(_LCD_CLEAR);                              // Limpia el LCD
Lcd_Out(1,1,latitudf);                            // Escribe la latitud en el LCD

//////////////////////////////////////          LONGITUD          ////////////////////////////////////////
memset(texto_longitud,0,15);                      // limpia los datos de
longitud...
memset(texto_grados ,0,15);                      // limpia los datos de
grados...
BuscarCampo(string,texto_longitud,5);             // Busca el campo de la
latitud
strncpy(texto_grados,texto_longitud,3);          // captura los grados
grados = atof(texto_grados);                     // FloatToStr(latitud,
texto_latitud);
Delay_us(20);                                     // Pausa de 20 us
minutos=atof(&texto_longitud[3])/60.0;           // apunta a los minutos
longitud=grados+minutos;                         // los pasa a grados

```

```

        BuscarCampo(string,texto_longitud,4);           // Busca el campo de la
coordenada E o W

        if (texto_longitud[0]=='S') longitud=-longitud; // Si apunta hacia el S
entonces sera negativa la latitud

        sprintf(texto_auxiliar,"Lon: %f",longitud);    //ACONDICIONA PARA EL LCD

        Lcd_Out(2,1,texto_auxiliar);                  // Escribe la longitud en el
LCD

        Delay_ms(2000);                               // Pausa de 2 seg.

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

//      Lcd_Chr_CP(' ');
//      Lcd_Chr_CP(string[30]);
//      Lcd_Chr_CP(' ');
//      Lcd_Chr_CP(string[44]);

} // end hora
//
// =====

// =====
//
//                      FUNCION PRINCIPAL
void main()
{
    configuracion_puertos();           // Configura los puertos
    inicio_lcd();                      // Inicia el LCD
    mensaje_inicio();                  // Muestra un Mensaje
    configuracion_interrupcion();      // Configura la interrupcion

    initSPI();                         // Inicia SPI a baja velocidad
    err = FAT32_Init();                // Inicia la micro SD

    if (err < 0) // comprueba si existe error
    {
        Lcd_Cmd(_LCD_CLEAR);          // Limpia LCD
        Lcd_Out(1,2,"Inicio Fallido"); // Envia mensaje
    }
}

```

```

PORTB.F4=1; // Enciende el led de aviso.
Delay_ms(1000); // Espera 1 segundo

while(err < 0) // ...reintentar cada segundo comprobando si sigue el error de
inicio
{
    err = FAT32_Init();
    Delay_ms(1000);
}
} // end err

initFastSPI(); // Inicia el SPI rápido
Lcd_Cmd(_LCD_CLEAR); // Limpia LCD
Lcd_Out(1,3,"Inicio OK"); // Envia mensaje: Inicio ok
PORTB = 0b10000000; // Enciende el led de aviso.
Delay_ms(1000); // Espera 1 segundo

// Crear una carpeta
err = 0; // err = 0;
entonces>>
if (err == FAT32_MakeDir("DIR_A")) // Create a new
folder: DIR_A
{
    Lcd_Cmd(_LCD_CLEAR); // Limpia LCD
    Lcd_Out(1,2,"Folder DIR_A"); // Envia mensaje
    Lcd_Out(2,5,"Creada"); // Envia mensaje
created
    Delay_ms(1000); // Espera 1 segundo
}
else
{
    Lcd_Cmd(_LCD_CLEAR); // Limpia LCD
    Lcd_Out(1,2,"Folder DIR_A"); // Envia mensaje
    Lcd_Out(2,3,"No Creada"); // Envia mensaje
    Delay_ms(2000); // Espera 1 segundo
}

//=====
//=====//
////////// BUCLE INFINITO
//////////

while (1)

```

```

{
RCSTA.F1 = 0; // OERR = 0
RCSTA.F2 = 0; // FERR = 0

if(listo == 1) // listo igual a 1?
{ // si...
    listo = 0; // listo = 0;
    velo = strstr(txt,"$GPVTG"); // Busca en la cadena la trama de $GPVTG
    if(velo != 0) // Si encuentra la identificacion $GPVTG,
    { // entonces,
        Velocidad_kmh(); // ejecuta la funcion Velocidad_kmh();
    } // end if/velo

    string = strstr(txt,"$GPRMC"); // Busca en la cadena la trama de $GPRMC
    if(string != 0) // Si encuentra la identificacion $GPRMC,
    { // entonces,
        Hora_Fecha(); // ejecuta la funcion Hora_Fecha();
        Latitud_Longitud(); // ejecuta la funcion Latitud_Longitud();
    } // end if/string
} // end if-listo

//=====//
//-----//
////////// GRABAR //////////

    err = FAT32_ChangeDir("DIR_A"); // Enter into DIR_A
Folder
    fhandle[0] = FAT32_Open("TEST.TXT", FILE_APPEND ); //open/create
TEXT.TXT

    if (fhandle >= 0)
    {
        Lcd_Cmd(_LCD_CLEAR); // Limpia LCD
        Lcd_Out(1,1,"Archivos Creados"); // Envia mensaje
        Lcd_Out(2,4,"Open OK"); // Envia mensaje
        Delay_ms(1000); // Espera 1 segundo
        Lcd_Cmd(_LCD_CLEAR); // Limpia LCD
        Lcd_Out(1,1,"Escribiendo..."); // Envia mensaje
        Delay_ms(1000); // Espera 1 segund
    }

```



```

//Write to files
FAT32_Write(fhhandle[0],"Velocidad: ",11);
FAT32_Write(fhhandle[0],texto_velo,      strlen(texto_velo));
FAT32_Write(fhhandle[0],"\\t", 1);
FAT32_Write(fhhandle[0],"Hora: ",7);
FAT32_Write(fhhandle[0],texto_hora,      strlen(texto_hora));
FAT32_Write(fhhandle[0],"\\t", 1);
FAT32_Write(fhhandle[0],"Fecha: ",8);
FAT32_Write(fhhandle[0],texto_fecha,      strlen(texto_fecha));
FAT32_Write(fhhandle[0],"\\t", 1);
//FAT32_Write(fhhandle[0],"Latitud: ",10);
FAT32_Write(fhhandle[0],latitudf,  strlen(latitudf));
FAT32_Write(fhhandle[0],"\\t\\t", 2);
//FAT32_Write(fhhandle[0],"Longitud: ",13);
FAT32_Write(fhhandle[0],texto_auxiliar,  strlen(texto_auxiliar));
FAT32_Write(fhhandle[0]," \\r\\r\\n\\n", 4);
Delay_ms(5000);                                     // Espera 1 segundo

// Cerrar todos los archivos
err = FAT32_Close(fhhandle[0]);                     // Cierra el archivo
TEXT.TXT
Delay_ms(500);                                     // Espera 1 segundo

if(err == 0)
{
    Lcd_Cmd(_LCD_CLEAR);                            // Limpia LCD
    Lcd_Out(1,2,"Texto escrito");                    // Envia mensaje
    Lcd_Out(2,4,"Correcto");                         // Envia mensaje
    Delay_ms(1000);                                  // Espera 1 segundo
}
else
{
    Lcd_Cmd(_LCD_CLEAR);                            // Limpia LCD
    Lcd_Out(1,4,"Archivos");                         // Envia mensaje
    Lcd_Out(2,3,"No Creados");                      // Envia mensaje
    Delay_ms(1000);                                  // Espera 1 segundo.
}

} // end while-main

```

ANEXO 2: Código Labview complete.

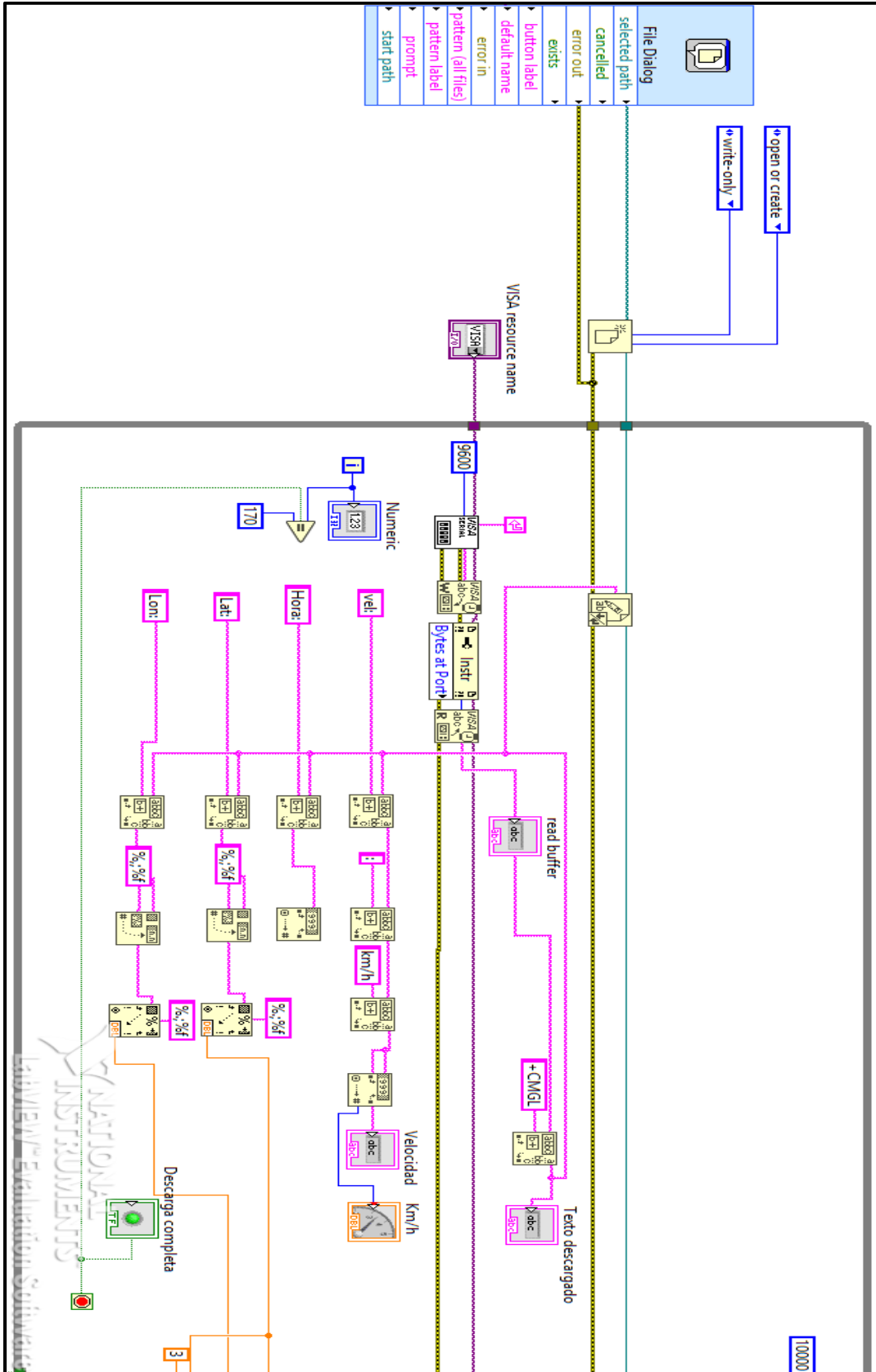


Figura 75. Código completo Labview (primera parte). Fuente: El autor.

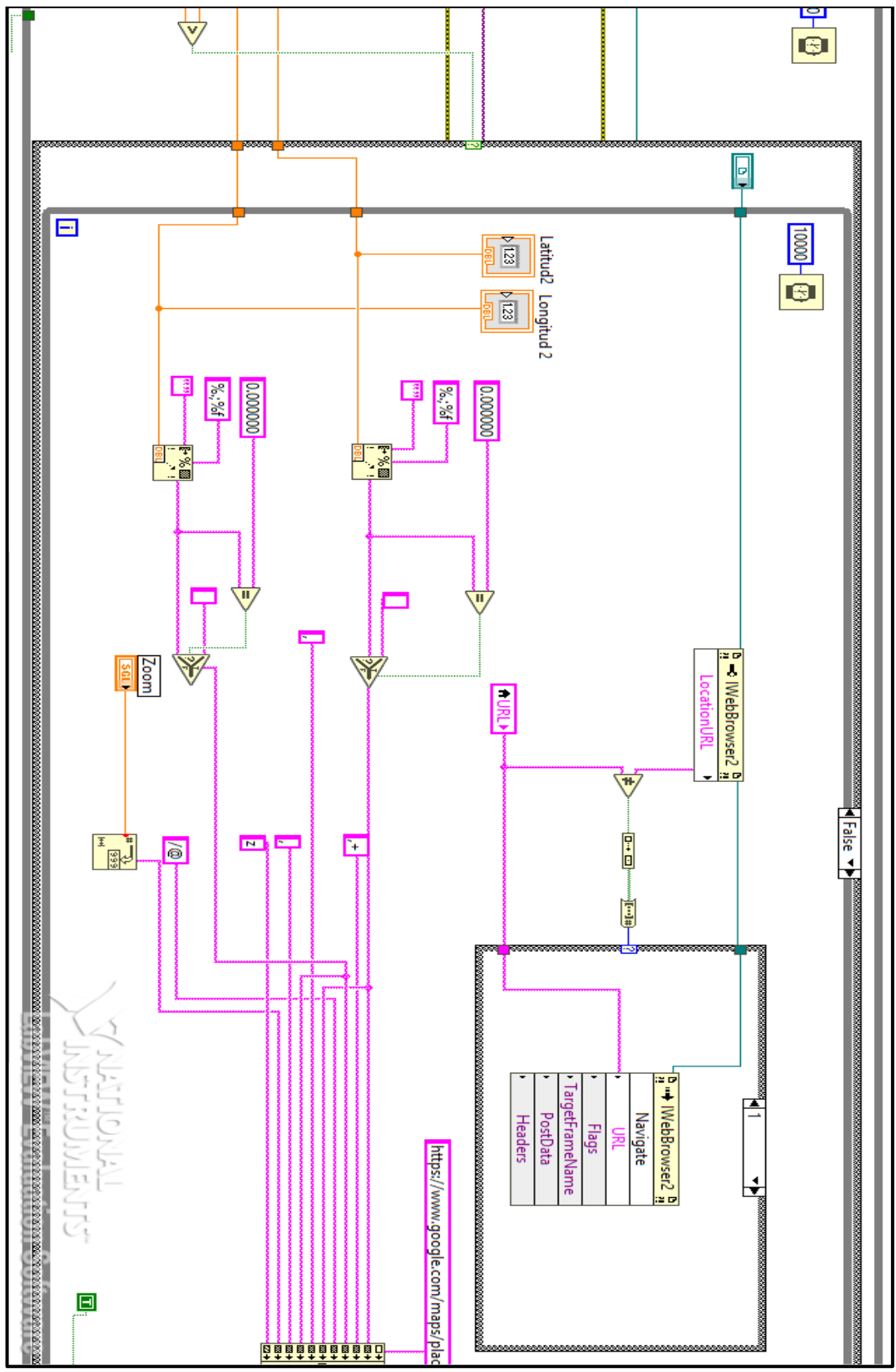


Figura 76. Código completo Labview (segunda parte). Fuente: El autor.

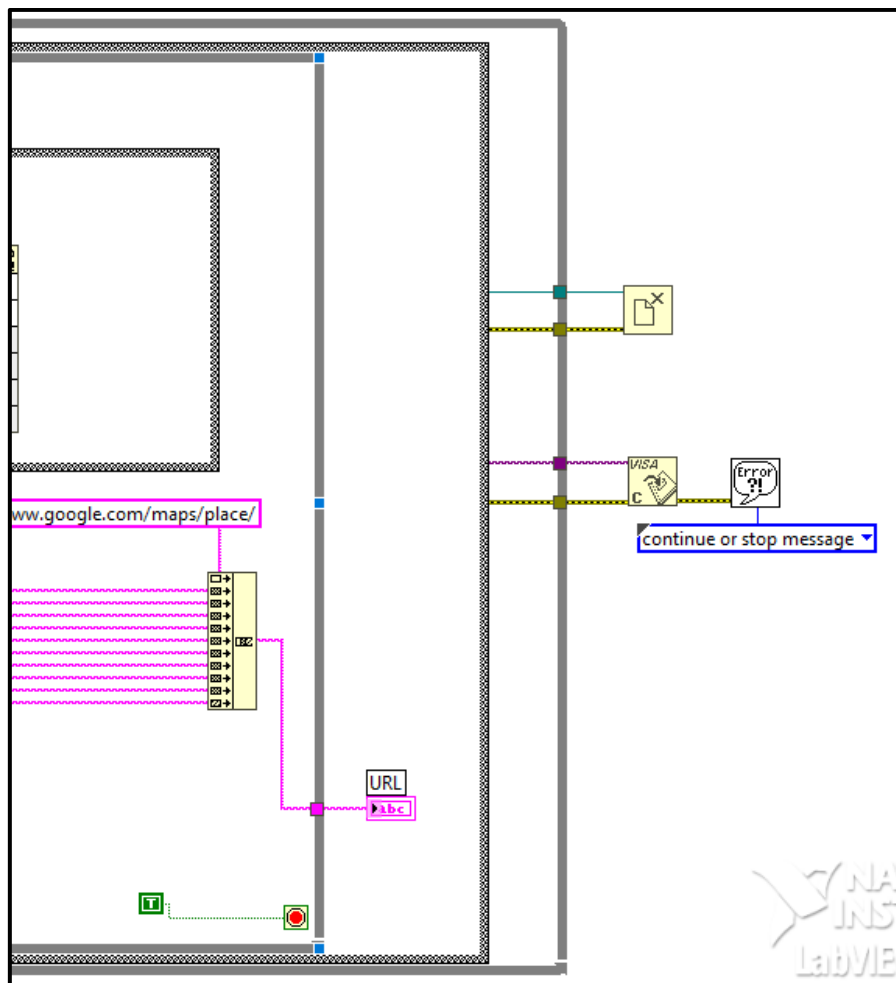


Figura 77-. Código completo Labview (tercera parte). Fuente: El autor.

ANEXO 3: Hojas de datos (SIM800L, GPS U-blox 6, PIB18F4620)



1. Introduction

This document describes SIM800L hardware interface in great detail.

This document can help user to quickly understand SIM800L interface specifications, electrical and mechanical details. With the help of this document and other SIM800L application notes, user guide, users can use SIM800L to design various applications quickly.

2. SIM800L Overview

SIM800L is a quad-band GSM/GPRS module, that works on frequencies GSM850MHz, EGSM900MHz, DCS1800MHz and PCS1900MHz. SIM800L features GPRS multi-slot class 12/ class 10 (optional) and supports the GPRS coding schemes CS-1, CS-2, CS-3 and CS-4.

With a tiny configuration of 15.8*17.8*2.4mm, SIM800L can meet almost all the space requirements in user applications, such as smart phone, PDA and other mobile devices.

SIM800L has 88pin pads of LGA packaging, and provides all hardware interfaces between the module and customers' boards.

- Support 5*5*2 keypads
- One full modem serial port, user can configure two serial ports
- One USB, the USB interfaces can debug, download software
- Audio channel which includes two microphone input; a receiver output and a speaker output
- Programmable general purpose input and output.
- A SIM card interface
- Support FM
- Support one PWM

SIM800L is designed with power saving technique so that the current consumption is as low as 0.7mA in sleep mode.

2.1. SIM800L Key Features

Table 1: SIM800L key features:

Feature	Implementation
Power supply	3.4V ~4.4V
Power saving	typical power consumption in sleep mode is 0.7mA (AT+CFUN=0)
Frequency bands	<ul style="list-style-type: none">● Quad-band: GSM 850, EGSM 900, DCS 1800, PCS 1900. SIM800L can search the 4 frequency bands automatically. The frequency bands can also be set by AT command "AT+CBAND". For details, please refer to document [1].● Compliant to GSM Phase 2/2+
Transmitting power	<ul style="list-style-type: none">● Class 4 (2W) at GSM 850 and EGSM 900● Class 1 (1W) at DCS 1800 and PCS 1900
GPRS connectivity	<ul style="list-style-type: none">● GPRS multi-slot class 12 (default)● GPRS multi-slot class 1~12 (option)
Temperature range	<ul style="list-style-type: none">● Normal operation: -40°C ~ +85°C

	<ul style="list-style-type: none"> Storage temperature -45°C ~ +90°C
Data GPRS	<ul style="list-style-type: none"> GPRS data downlink transfer: max. 85.6 kbps GPRS data uplink transfer: max. 85.6 kbps Coding scheme: CS-1, CS-2, CS-3 and CS-4 PAP protocol for PPP connect Integrate the TCP/IP protocol. Support Packet Broadcast Control Channel (PBCCH) CSD transmission rates: 2.4, 4.8, 9.6, 14.4 kbps
CSD	<ul style="list-style-type: none"> Support CSD transmission
USSD	<ul style="list-style-type: none"> Unstructured Supplementary Services Data (USSD) support
SMS	<ul style="list-style-type: none"> MT, MO, CB, Text and PDU mode SMS storage: SIM card
SIM interface	Support SIM card: 1.8V, 3V
External antenna	Antenna pad
Audio features	Speech codec modes: <ul style="list-style-type: none"> Half Rate (ETS 06.20) Full Rate (ETS 06.10) Enhanced Full Rate (ETS 06.50 / 06.60 / 06.80) Adaptive multi rate (AMR) Echo Cancellation Noise Suppression
Serial port and debug port	Serial port: <ul style="list-style-type: none"> Full modem interface with status and control lines, unbalanced, asynchronous. 1200bps to 115200bps. Can be used for AT commands or data stream. Support RTS/CTS hardware handshake and software ON/OFF flow control. Multiplex ability according to GSM 07.10 Multiplexer Protocol. Autobauding supports baud rate from 1200 bps to 57600bps. upgrading firmware Debug port: <ul style="list-style-type: none"> USB_DM and USB_DP Can be used for debugging and upgrading firmware.
Phonebook management	Support phonebook types: SM, FD, LD, RC, ON, MC.
SIM application toolkit	GSM 11.14 Release 99
Real time clock	Support RTC
Timing functions	Use AT command set
Physical characteristics	Size:15.8*17.8*2.4mm Weight:1.35g
Firmware upgrade	Main serial port or USB port.

1.3 GPS performance

Parameter	Specification			
Receiver type	50 Channels GPS L1 frequency, C/A Code SBAS: WAAS, EGNOS, MSAS			
Time-To-First-Fix ¹		NEO-6G/Q/T	NEO-6MV	NEO-6P
	Cold Start ¹	26 s	27 s	32 s
	Warm Start ²	26 s	27 s	32 s
	Hot Start ¹	1 s	1 s	1 s
	Aided Starts ³	1 s	<3 s	<3 s
Sensitivity ⁴		NEO-6G/Q/T	NEO-6MV	NEO-6P
	Tracking & Navigation	-162 dBm	-161 dBm	-160 dBm
	Reacquisition ⁵	-160 dBm	-160 dBm	-160 dBm
	Cold Start (without aiding)	-148 dBm	-147 dBm	-146 dBm
	Hot Start	-157 dBm	-156 dBm	-155 dBm
Maximum Navigation update rate		NEO-6G/Q/MT	NEO-6PV	
		5Hz	1 Hz	
Horizontal position accuracy ⁶	GPS	2.5 m		
	SBAS	2.0 m		
	SBAS + PPP ⁷	< 1 m (2D, R50) ⁸		
	SBAS + PPP ¹	< 2 m (3D, R50) ⁸		
Configurable Timepulse frequency range		NEO-6G/Q/MPV	NEO-6T	
		0.25 Hz to 1 kHz	0.25 Hz to 10 MHz	
Accuracy for Timepulse signal	RMS	30 ns		
	99%	<60 ns		
	Granularity	21 ns		
	Compensated ⁹	15 ns		
Velocity accuracy ⁶		0.1 m/s		
Heading accuracy ⁶		0.5 degrees		
Operational Limits	Dynamics	≤ 4 g		
	Altitude ¹⁰	50,000 m		
	Velocity ¹⁰	500 m/s		

Table 2: NEO-6 GPS performance

PIC18F2525/2620/4525/4620

TABLE 1-1: DEVICE FEATURES

Features	PIC18F2525	PIC18F2620	PIC18F4525	PIC18F4620
Operating Frequency	DC – 40 MHz	DC – 40 MHz	DC – 40 MHz	DC – 40 MHz
Program Memory (Bytes)	49152	65536	49152	65536
Program Memory (Instructions)	24576	32768	24576	32768
Data Memory (Bytes)	3968	3968	3968	3968
Data EEPROM Memory (Bytes)	1024	1024	1024	1024
Interrupt Sources	19	19	20	20
I/O Ports	Ports A, B, C, (E)	Ports A, B, C, (E)	Ports A, B, C, D, E	Ports A, B, C, D, E
Timers	4	4	4	4
Capture/Compare/PWM Modules	2	2	1	1
Enhanced Capture/Compare/ PWM Modules	0	0	1	1
Serial Communications	MSSP, Enhanced USART	MSSP, Enhanced USART	MSSP, Enhanced USART	MSSP, Enhanced USART
Parallel Communications (PSP)	No	No	Yes	Yes
10-Bit Analog-to-Digital Module	10 Input Channels	10 Input Channels	13 Input Channels	13 Input Channels
Resets (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT
Programmable Low-Voltage Detect	Yes	Yes	Yes	Yes
Programmable Brown-out Reset	Yes	Yes	Yes	Yes
Instruction Set	75 Instructions; 83 with Extended Instruction Set Enabled	75 Instructions; 83 with Extended Instruction Set Enabled	75 Instructions; 83 with Extended Instruction Set Enabled	75 Instructions; 83 with Extended Instruction Set Enabled
Packages	28-Pin SPDIP 28-Pin SOIC	28-Pin SPDIP 28-Pin SOIC	40-Pin PDIP 44-Pin QFN 44-Pin TQFP	40-Pin PDIP 44-Pin QFN 44-Pin TQFP

ANEXO 4: Solicitud de autorización dirigida a la ANT (Agencia Nacional de Transito)