

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

IMPLEMENTACIÓN DE UN PROTOTIPO DE VELOCÍMETRO QUE NOTIFIQUE EXCESOS DE VELOCIDAD VÍA SMS

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO
SUPERIOR EN REDES Y TELECOMUNICACIONES**

JUAN CARLOS VALDEZ BETANCOURT

juan.valdez@epn.edu.ec

DIRECTOR: ING. CARLOS ANDRES YUNGA SANCHEZ

carlos.yunga@epn.edu.ec

DMQ, febrero 2024

CERTIFICACIONES

Yo, JUAN CARLOS VALDEZ BETANCOURT declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

JUAN CARLOS VALDEZ BETANCOURT

juan.valdez@epn.edu.ec

juank-surdo@hotmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por JUAN CARLOS VALDEZ BETANCOURT, bajo mi supervisión.

ING. CARLOS ANDRES YUNGA SANCHEZ

DIRECTOR

carlos.yunga@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

Juan Carlos Valdez Betancourt

DEDICATORIA

El presente trabajo va dedicado a la Escuela Politécnica Nacional por brindarme la oportunidad de realizar mis estudios dentro de sus instalaciones, así como darme los recursos necesarios para mejorar mis conocimientos para un entorno laboral.

Dedicado también a mis padres que día a día hacen el esfuerzo por sacar adelante tanto para mi persona como a mis hermanos, así se encuentren enfermos siempre vieron por el bienestar de todos.

AGRADECIMIENTO

Quiero empezar agradeciendo a Dios y la Virgen María por darme la sabiduría para seguir un camino lleno de bendiciones.

A mis padres por brindarme el apoyo, la confianza, el amor, cariño en el transcurso de mi vida, también por los consejos, las charlas y valores que me han inculcado para ser la persona que soy ahora, a mis hermanos por brindarme la confianza de seguir adelante y la compañía en noches de estudio.

Agradezco al resto de mi familia que estuvieron brindándome el apoyo necesario y dándome consejos para cumplir mis metas y propósitos.

A mis maestros que impartieron los conocimientos necesarios en mi para mi formación estudiantil y el apoyo para salir adelante.

Finalmente, agradezco al ingeniero Carlos Yunga que tuvo la paciencia suficiente en el proceso de elaboración del proyecto de integración y la ayuda de la creación del mismo.

ÍNDICE DE CONTENIDOS

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDOS.....	V
RESUMEN	VII
ABSTRACT	VIII
1. DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general.....	2
1.2 Objetivos específicos	2
1.3 Alcance	2
1.4 Marco teórico.....	2
Arduino.....	2
LCD	3
GSM.....	3
Buzzer	4
Proteus	4
2. METODOLOGÍA.....	4
3. RESULTADOS	5
3.1 Identificar los componentes de hardware y software necesario para la implementación del velocímetro	6
Componentes de software	6
Componentes de hardware.....	6
3.2 Diseñar el prototipo del velocímetro.....	8
Principio de funcionamiento del velocímetro	8
Creación código de programación del velocímetro	9
Armado del circuito del prototipo de velocímetro	14
3.3 Implementación del prototipo del velocímetro	15
Creación PCB para el prototipo de velocímetro	15
Implementación de cubierta para el prototipo de velocímetro	20
Instalación del prototipo de velocímetro en un vehículo	21
3.4 Realizar pruebas de funcionamiento del prototipo.....	22
4. CONCLUSIONES	25
5. RECOMENDACIONES	27
6. REFERENCIAS BIBLIOGRÁFICAS.....	28

Referencias	28
7. ANEXOS	I
ANEXO I: Certificado de Originalidad.....	I
ANEXO II: Enlaces	II
ANEXO III: Código Fuente.....	III

RESUMEN

En el siguiente documento se presenta un proyecto que tiene como objetivo la implementación de un prototipo portátil de velocímetro que detecte excesos de velocidad y envíe un mensaje de alerta con una activación de un buzzer (zumbador), el cual estará controlado por un microcontrolador de Arduino, modulo GPS, GSM y pantalla LCD, adicional contará con una fuente de alimentación propia y un método de carga por cable USB.

Dentro de la primera sección se especifica los objetivos tanto general como específicos del prototipo, para establecer las necesidades requeridas que deba cumplir el velocímetro, también se detalla el alcance que tendrá el prototipo y las características principales de los elementos que estarán conectados.

En la segunda sección se presenta la metodología del proyecto donde se tiene una breve descripción de los objetivos específicos que contiene el prototipo de velocímetro, así como un breve análisis de los elementos.

La tercera sección se detalla los resultados obtenidos del velocímetro, así como una descripción detallada de los objetivos específicos y de los elementos que conforman el proyecto y se especifica paso a paso como se realizó el prototipo.

Finalmente, en las últimas secciones se tienen las conclusiones y recomendaciones que se obtuvieron al realizar el prototipo, también se da un análisis de los resultados en comparación a los resultados esperados.

PALABRAS CLAVE: Velocímetro, GPS, GSM, Arduino, buzzer, LCD

ABSTRACT

The following document presents a project that aims to implement a portable speedometer prototype that detects speeding and sends an alert message with a buzzer activation, which will be controlled by an Arduino microcontroller, GPS module, GSM, and LCD display, in addition it will have its own power supply and a charging method via USB cable.

The first section specifies both the general and specific objectives of the prototype, to establish the required needs to be met by the speedometer, also details the scope of the prototype and the main characteristics of the elements that will be connected.

The second section presents the methodology of the project where there is a brief description of the specific objectives of the speedometer prototype, as well as a brief analysis of the elements.

The third section details the results obtained from the speedometer, as well as a detailed description of the specific objectives and the elements that make up the project and specifies step by step how the prototype was made.

Finally, the last sections contain the conclusions and recommendations obtained from the prototype, as well as an analysis of the results compared to the expected results.

KEY WORDS: *Speedometer, GPS, GSM, Arduino, buzzer, LCD.*

1. DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

En el presente proyecto de titulación se realizará el diseño e implementación de un prototipo de velocímetro que permita notificar cuando un vehículo ha excedido cierto rango de velocidad. Esto permitirá tener un control sobre la velocidad en que circula un determinado vehículo y tomar acciones preventivas para evitar accidentes.

A medida que existen nuevos modelos de automóviles con muchas mejoras que sus versiones pasadas, los seres humanos han optado por incrementar la velocidad dentro de carreteras, esto conlleva a trágicos desenlaces tanto para las personas que conducen a alta velocidad como para personas que no tuvieron nada que ver o incluso daños en infraestructuras, por eso se optó por un prototipo de velocímetro que muestre por mensaje una alerta de exceso de velocidad con la ubicación del prototipo y simultáneamente la activación de un *buzzer* (zumbador), que se desactive cuando ya no exista excesos de velocidad.

Este prototipo al ser portátil brindará un rendimiento óptimo de energía, esto se debe a la implementación de baterías de litio y un mecanismo para poder cargarlas, las baterías de litio poseen grandes ciclos de vida y son usadas para la fuente de alimentación, además cuentan con propiedades de no sobrecalentarse y tampoco son inflamables, lo que brinda una mayor seguridad en el lugar instalado como a las personas que lo manipulen.

Con el uso del módulo GPS Neo7M se tendrá cobertura en cualquier parte del mundo debido a que recibe datos de los satélites GEO, LEO y MEO. Los satélites GEO dan una cobertura total de la Tierra debido a su distancia de separación del planeta, en cambio a los satélites MEO se encuentran con una separación menor a los GEO, pero pueden transmitir con menos latencias y una mayor velocidad de transmisión, comúnmente el módulo Neo7M usa los datos que se obtienen de estos satélites por su velocidad de transmisión y sus coberturas en latitudes extremas.

El módulo GSM puede operarse en todo el mundo para enviar mensajes, por las características que contiene, ya que puede operar en las bandas de frecuencias usadas para redes GSM, el módulo GSM al conllevar las diferentes frecuencias puede usar chip de cualquier operadora telefónica disponible.

1.1 Objetivo general

Implementar un prototipo de velocímetro con notificaciones de exceso de velocidad vía SMS.

1.2 Objetivos específicos

- Identificar los componentes de hardware y software necesarios para la implementación del proyecto.
- Diseñar el prototipo.
- Implementar el prototipo.
- Realizar pruebas de funcionamiento.

1.3 Alcance

En el presente proyecto se implementará un prototipo de velocímetro portátil que pueda ubicarse dentro de un vehículo. El prototipo debe enviar alertas de velocidad vía SMS.

La Agencia Nacional de Tránsito establece tres tipos de velocidades según el vehículo que se esté conduciendo. El prototipo debe tener un sistema (aplicación) remoto que permita al administrador seleccionar el vehículo que se está conduciendo y en qué zona (Urbana, perimetral o Carretera) se encuentra para determinar la máxima velocidad a la que puede circular.

Si se supera el límite permitido se debe enviar un SMS al administrador con la velocidad y la ubicación del vehículo. Adicional el prototipo debe tener un buzzer el cual alerte al conductor que ha excedido la velocidad.

El prototipo debe tener una fuente de alimentación propia y adicional que se pueda conectar para cargarse mediante USB.

Hay que considerar que el prototipo no será funcional dentro de túneles de grandes dimensiones.

1.4 Marco teórico

Arduino

Arduino es una plataforma encargada de crear circuitos electrónicos con base en código abierto y ayuda de software y hardware libre, la plataforma de Arduino permite

crear proyectos con base en diferentes necesidades posibles, poseen un elemento centro que se encarga de la comunicación con otros circuitos o elementos conectados, este elemento se denomina microcontrolador que permite conectar entre sus diferentes periféricos tanto de entrada como de salida, obteniendo los datos de Arduino para darles nuevas funciones o mejoras de la misma placa [1].

Arduino es una placa PCB (*"Printed Circuit Board"*), es decir, es una placa de circuito impreso fabricada con materiales no conductores con capas de material conductor para su circuitería, la circuitería que tiene la placa Arduino cuenta con todas las necesidades para establecer la comunicación entre los elementos internos de la placa y para desarrollar las diferentes aplicaciones que se programen al realizar un proyecto, a pesar que existen diferentes tipos de modelos, todas comparten la mayor parte de sus características de arquitectura y librerías usadas con otros modelos [2].

LCD

La LCD (*"Liquid Crystal Display"*) es un dispositivo electrónico que consta de una pantalla de cristal líquida, que se usa comúnmente para visualizar imágenes o datos recibidos desde un microcontrolador o cualquier dispositivo que envíe datos para ser visualizados, se pueden clasificar dependiendo de la cantidad de píxeles agrupados para formar la pantalla, para la formación de colores y movimientos de las imágenes los electrodos que conforman las moléculas de cristal líquido. Los electrodos se tuercen y se mueven dependiendo de la carga eléctrica suministrada a la pantalla [3].

GSM

GSM (*"Global System for Mobile Communications"*) corresponde al acrónimo de Sistema Global para las Comunicaciones Móviles y se trata de tener conexiones mediante ondas de radiofrecuencia, las conexiones de este sistema son inalámbricas y se transmiten por medio de células radiales [4].

La arquitectura se basa en dispositivos móviles o terminal, las estaciones base (encargadas de dar la comunicación inalámbrica por medio de las antenas que dan cobertura para la conexión) y las tarjetas SIM (*"Subscriber Identify Module"*) son las que contienen los detalles del dispositivo del usuario, por otro lado, se tiene los controladores de las respectivas estaciones base encargadas de las gestiones de los recursos que tendrá cada estación base, como último elemento de la arquitectura se tiene el centro de conmutación donde se recopila los datos y se comprueba la identidad de la tarjeta SIM, para utilizar los servicios de la red conectada [4].

Las redes GSM usan diferentes frecuencias del espectro radioeléctrico dependiendo de la zona a la que esté conectada, un ejemplo es en Europa que usa frecuencias de 900 y 1800 MHz, en América es distinto ya que usa la frecuencia de 1900 MHz, esto conlleva a un gran problema para poder usar equipos en todo el mundo debido a sus diferentes frecuencias de trabajo para ofrecer sus servicios, las principales utilidades de redes GSM se encuentran en el envío y recepción de mensajes, llamadas y la conexión a Internet para el uso de datos [5].

Buzzer

Un buzzer es un pequeño dispositivo que se encarga de realizar conversiones de señal eléctrica a sonido, estos dispositivos generan un audio aceptable usados para proyectos específicos donde se requiera una acción necesaria de los buzzer, como pueden ser alarmas o controles de sonido para electrodomésticos, el funcionamiento de los buzzer depende de la fuente de alimentación suministrada al dispositivo [6].

Proteus

Proteus 8 es un software que se encarga de la simulación de circuitos electrónicos y creación de circuitos impresos, mejor conocido como PCB, comúnmente es usado para la creación de circuitos digitales impresos, esto se debe a la gran cantidad de herramientas usadas que cuenta, como lo es el auto routing para la creación automática de pistas, la posibilidad de modificar el tamaño de las pistas, el visualizador digital de los circuitos, entre otras herramientas, otra de las aplicaciones usadas de Proteus es los circuitos electrónicos, que ayuda a comprender el funcionamiento de elementos y dispositivos electrónicos, así como configurarlos [7].

2. METODOLOGÍA

Como etapa inicial se realizó el respectivo análisis del proyecto, para así comprender los objetivos y el alcance que tiene el mismo, para definir las características principales y necesarias para cumplir con los requerimientos establecidos del respectivo proyecto.

Al tener en claro las características del proyecto, se analizan los elementos necesarios para cumplir de manera eficiente las necesidades propuestas, entre los elementos a definir se tiene el modelo de la placa Arduino, módulo GPS, módulo GSM y la pantalla donde se evidenciarán los datos de velocidad que alcanza el vehículo, y el respectivo análisis de la velocidad dependiendo de la zona para determinar el límite de velocidad que puede alcanzar, a su vez se implementara el código para el control

de los dispositivos y envió de los mensajes de alerta.

Al analizar las características que el prototipo va a tener, se crea el código del programa con el que estará cargada la placa de Arduino, como son, las funciones creadas para la comunicación entre placa Arduino y placa GSM, las órdenes para la obtención de los diferentes datos, las respectivas instrucciones para el control de la obtención de los datos de velocidad y ubicación GPS del vehículo, también se tendrá el respectivo código para el envío de mensaje y el activador del buzzer al sobrepasar cierta velocidad dependiendo de la zona escogida.

Analizando los elementos necesarios del proyecto, se realiza el respectivo armado del prototipo del velocímetro con base en los elementos ya seleccionados y el código creado para él. Se toman los datos obtenidos para comprobar si el código funciona bien y operan correctamente los elementos escogidos, si no operan correctamente, se analiza si hay error en el código o en los elementos seleccionados, para modificar o cambiar donde exista el error.

Como siguiente punto se arma la placa PCB, desarrollando las pistas y líneas de conexión entre los elementos, para ello se debe conseguir una fuente de alimentación que se encargara de suministrar la energía necesaria para todo el circuito y el elemento necesario para realizar la carga de la fuente de alimentación mediante USB, al obtener listas las placas PCB con todos los elementos colocados y soldados, se instala dentro del vehículo para realizar las pruebas de funcionamiento con baja, mediana y alta velocidad.

3. RESULTADOS

En esta sección se mostrarán los resultados obtenidos de cada objetivo propuesto, así como las características que se tienen para el desarrollo efectivo de los diferentes objetivos, entre ellos, la identificación de los componentes de hardware y software requeridos, el diseño del prototipo, la implementación del mismo y las pruebas de funcionamiento para analizar la velocidad obtenida y envió de mensaje de alerta.

3.1 Identificar los componentes de hardware y software necesario para la implementación del velocímetro

Dentro de los requerimientos propuestos del prototipo de velocímetro, se optó por utilizar los siguientes elementos para obtener los mejores resultados y de manera eficiente cumpliendo las necesidades requeridas:

Componentes de software

Para la creación del código a utilizar dentro del prototipo del velocímetro se optó por la plataforma Arduino IDE, la cual se encargará de establecer el código de ejecución de las diferentes funciones, reglas, la declaración de variables, el texto del mensaje de alerta, entre otras funcionalidades que se tendrá en el prototipo, como punto adicional se tiene que el programa Arduino IDE permite seleccionar si se desea programar en otro tipo de placa diferente a la de Arduino para cargar el código que se crea.

Para la creación de la placa PCB se usa el programa Ares, lo que brinda herramientas para su creación del circuito digital y brinda el respectivo conocimiento de cómo irán conectados cada elemento del prototipo, es decir, la estructura del prototipo una vez finalizado.

Componentes de hardware

Para la elección del hardware requerido como microcontrolador necesario para cumplir las necesidades óptimas para el prototipo, se obtuvo entre dos opciones que son Arduino Uno y Esp32, de los cuales se escogió entre la primera opción debido a que cuenta con características óptimas y tiene mejor conectividad con el módulo GSM, aunque una placa Esp32 cuenta con mejores funcionalidades y características que Arduino, como son las diferentes memorias que posee cada placa, la cantidad de pines que poseen tanto digitales como analógicos son mayores y la integración de shields ya definidos [8].

Las placas Arduino poseen mejores características en sus pines analógicos y se puede visualizar todos los datos de la entrada analógica sin usar multiplexores, como es el caso de las placas Esp32, además se tiene mayores compatibilidades con una amplia gama de *shields* y una mayor facilidad de uso con la documentación disponible, como se puede observar en la **Tabla 3.1** se detalla las principales características comparativas entre las dos placas [8].

Tabla 3.1. Comparación Esp32 y Arduino UNO [8]

Placas Características	Esp32	Arduino UNO
Voltaje de funcionamiento	3.3 [V]	5 [V]
Consumo de energía	20 – 240 [mA]	45 – 80 [mA]
Pines de entrada y salida analógicos	36	14
Pines entrada y salidas digitales	15	6
Memoria Flash	4 MB	32 KB
Memoria SRAM	520 KB	2KB

Por parte del módulo GPS se tiene el módulo Neo 7Mv2, es comúnmente usado en aplicaciones de geolocalización, la recepción de datos de posición se lo realiza por medio de su antena y enviadas al microcontrolador para ser tratadas, los datos obtenidos también son guardados dentro de la memoria interna del módulo, cuenta también con una batería usada para respaldar las configuraciones del GPS al recibir información de los respectivos satélites que transitan en la órbita terrestre, su alimentación esta entre los 2.7 [V] a 3.6 [V] a una corriente de 37 [mA], aunque también puede ser alimentada a 5 [V] [9].

La placa usada para el envío de mensajes es el módulo SIM808 con un chip de la operadora de servicios de “Claro”, esto da la opción de tener servicios datos para navegar en Internet, servicios móviles para el envío de los respectivos mensajes de alerta, el módulo SIM808 permite la adaptación de servicios GPS, GSM el cual pueden ser ofrecidos por medio de sus adaptadores para antenas de los diferentes servicios, para el uso del servicio GSM el módulo puede operar dentro de cualquier parte del mundo, debido a que opera dentro de las cuatro bandas de frecuencias disponibles para el uso del estándar GSM, que son 850, 900, 1800 y 1900 MHz, puede operar con redes de 2G, dentro de Ecuador las bandas de frecuencias usadas para su operación en redes móviles son 850 MHz y 1900 MHz y puede ser compatible con la cualquier operadora del país [10].

Además de enviar mensajes el módulo permite recibir los mismos para la comunicación entre dos dispositivos, entre otra de las opciones del módulo GSM se tiene el SIM900, al operar en las mismas bandas de frecuencia puede ser óptimo para el uso en el prototipo, pero sus características no satisfacen las necesidades al momento de conectarla y hacerle operar con su respectiva antena GPS, en la **Tabla 3.2** se pueden observar las características de los módulos GSM pronunciados [10].

Tabla 3.2. Comparación SIM808 y SIM900 [10]

Módulos	SIM808	SIM900
GSM Características		
Voltaje de funcionamiento	3,3 – 5 [V]	9 – 20 [V]
Consumo de energía	2000 [mA]	1,5 – 2 [mA]
Bandas de frecuencia usadas	850/900/1800/1900 MHz	850/900/1800/1900 MHz

Para la visualización de los datos de velocidad que se obtiene del módulo GPS Neo7M, se los puede presentar por medio de una pantalla LCD 16x2, esto quiere decir que se pueden tener hasta 32 caracteres siendo distribuidos en dos filas, donde en la primera fila cuenta con los primeros 16 caracteres y la segunda con los 16 faltantes, el uso en el prototipo es de forma que en la primera fila se presentara el mensaje de velocidad y en la segunda los datos de velocidad que se obtiene por medio del módulo GPS, cuando el GPS no se encuentre conectado o no se reciba señales del mismo en la pantalla se visualizara "INVALID" dentro de la primera fila [11].

Como último elemento es la fuente de alimentación que será usada para las pruebas de funcionamiento, la fuente que se usa es por medio de baterías de ion litio del tipo LifePO4, las cuales presentan las características de ser recargables, son usadas comúnmente para proyectos que conlleven alta capacidad de consumo de energía, operan en un voltaje de 3,6V, en el caso del prototipo son usadas para tener una fuente de alimentación con un gran ciclo de vida útil, cuenta con la característica de no sobrecalentarse y si existe algún tipo de perforación en las baterías estas no llegan a inflamarse, en el prototipo las baterías contarán con un mecanismo para recargarse por medio de un conector USB [12].

La fuente de alimentación contará con dos baterías de litio, las dos baterías de litio se encuentran conectadas en paralelo para dar una mayor capacidad de corriente y energía, a su vez de dar una mayor protección contra el sobrecalentamiento de cada una de las baterías, la fuente de alimentación también cuenta con un convertidor a 5 [V] para suministrar a los elementos del prototipo sin que ocurra sobretensiones hacia cada elemento, el método de carga para las baterías de litio es mediante el porta pilas que cuenta con conexión por medio de cable a USB, a su vez posee un sistema de protección ante cortocircuitos, al ocurrir un cortocircuito el prototipo se reiniciara almacenando los datos que se tenían antes de su reinicio.

3.2 Diseñar el prototipo del velocímetro

Principio de funcionamiento del velocímetro

Para comprender el principio de funcionamiento del velocímetro se debe tener en cuenta la forma en la que estará comunicándose los respectivos elementos del prototipo y como estarán transmitiendo sus datos entre ellos, como se puede ver en la **Figura 3.1** se tiene el principio básico de la respectiva comunicación de los elementos, el elemento encargado de recibir todos los datos es el módulo GPS, el cual le envía los mismos a la placa de Arduino UNO para posteriormente ser transferidos a la pantalla LCD donde se mostrara la respectiva velocidad.

Finalmente, el módulo GSM SIM808 enviara la respectiva notificación de alerta y la ubicación donde se encuentra el prototipo por medio de un mensaje de texto al número agregado en el código de programación y a su vez al respectivo buzzer que estará encargado de dar la alerta de disminuir la velocidad por el sonido que se activara cuando se sobrepase la velocidad configurada en el código de programación.

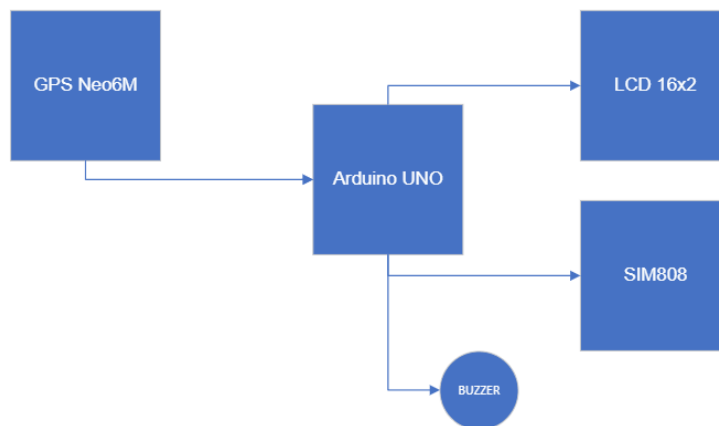


Figura 3.1. Principio de funcionamiento

Creación código de programación del velocímetro

Conociendo el principio de funcionamiento del velocímetro se procede a realizar el código respectivo para el control y funcionamiento del velocímetro, el primer paso que se realiza es insertar las respectivas librerías de los elementos usados, el siguiente paso es describir las variables necesarias para el código, como se puede ver en la **Figura 3.2** se tiene el tipo de variables y los valores que se almacenaran en ellos.

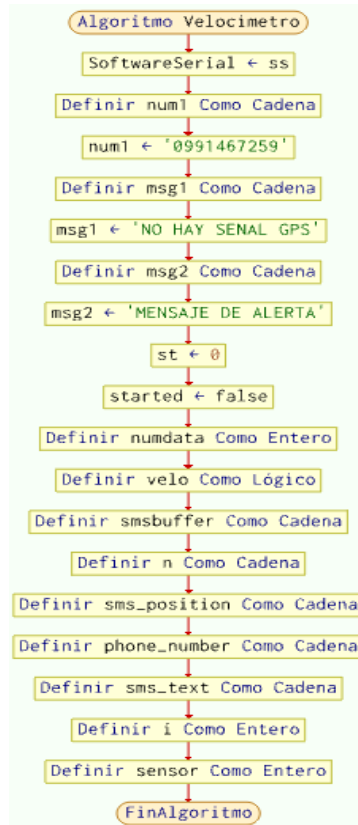


Figura 3.2. Declaración de variables

Al tener las variables necesarias y las librerías que se usaran en el código, se procede a realizar la configuración de los elementos necesarios dentro de la función “void setup”, estas configuraciones son para la pantalla LCD y la configuración del mensaje inicial que tendrá la pantalla al tener los datos del módulo GSM, como se observa en la Figura 3.3 se tiene el diagrama de flujo de la función.

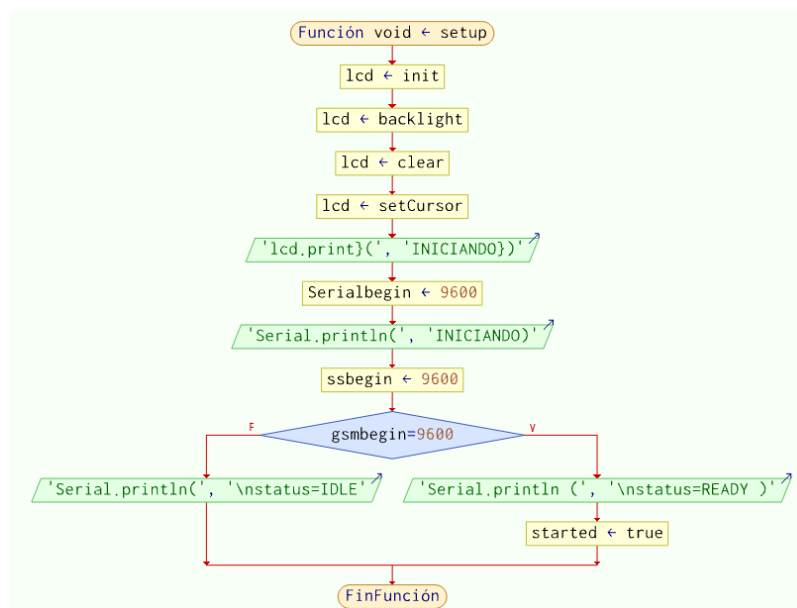


Figura 3.3. Void Setup

Dentro de la función “void loop” se insertan los comandos necesarios para ejecutarse dentro del programa y posteriormente a los elementos conectados del prototipo, como se observa en la **Figura 3.4** se tiene la sentencia “si” que servirá como mecanismo de límite de la velocidad de un vehículo, cuando el valor de la sentencia da un resultado verdadero se llama a una nueva función que servirá para él envió del mensaje y la llamada a la función de la lectura de los datos del GPS.

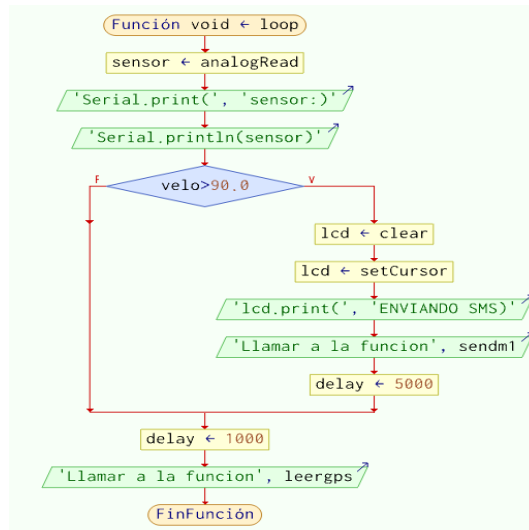


Figura 3.4. Void Loop

Para la lectura de los datos del GPS se crea una nueva función conocida con “void leergps”, que estará encargado de obtener los datos del módulo GPS para transmitirlos hacia la placa Arduino y posteriormente a la pantalla LCD, donde un inicio solo se mostrara la pantalla en blanco hasta tener al módulo GPS activo, como se observa en la **Figura 3.5** se tiene el diagrama de la respectiva función.

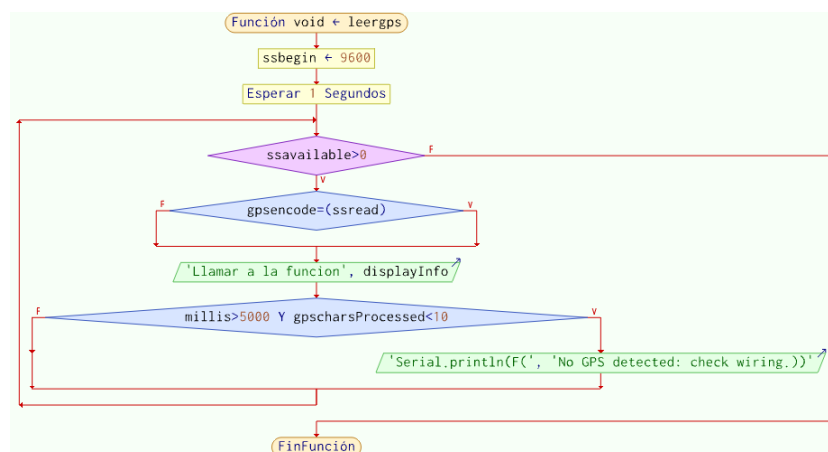


Figura 3.5. Void Leergps

Dentro de la configuración de la pantalla LCD se tendrá la siguiente configuración como se muestra en la **Figura 3.6** el cual muestra la sentencia “si” donde al obtener un valor verdadero al programado se crearan las variables de latitud y longitud de la función, donde se almacenarán los respectivos datos obtenidos respectivamente del

módulo GPS y los datos de velocidad que se obtienen.

El siguiente punto que se programa es él envió del mensaje de la ubicación con los datos de latitud y longitud previamente obtenidos, para enviar la latitud y longitud por medio del enlace URL para “Google maps” se debe transformar los valores de lógicos a caracteres.

Con la velocidad obtenida, se almacena dentro de la variable creada para dicha acción dentro de la función y que se transmitirá a la pantalla LCD, esta acción se ejecutará siempre que el vehículo se haya desplazado o en movimiento.

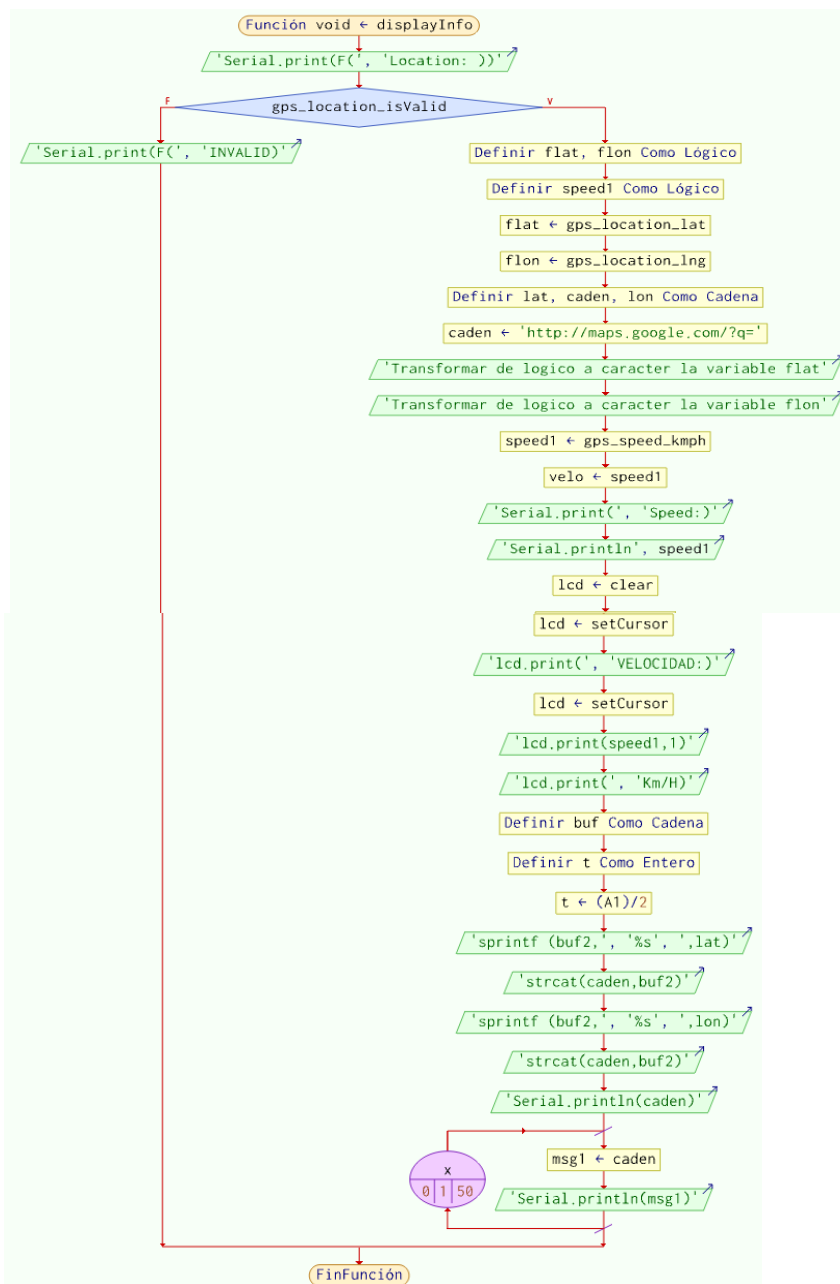


Figura 3.6. Void Displayinfo

La siguiente función creada se la conoce como “*void recunread*”, que se encargara de analizar los datos para el respectivo envío del mensaje de texto, como se observa en la **Figura 3.7** se tiene la configuración necesaria para recopilar la información para posteriormente ser enviada a una nueva función la cual comparara los datos obtenidos con los válidos y que acción tomar con dichos datos.

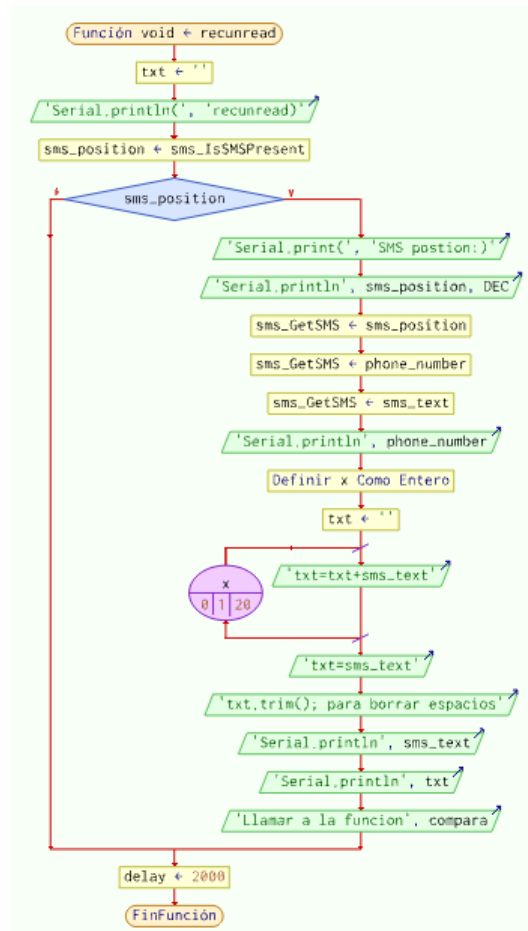


Figura 3.7. Void Recunread

La función encargada de realizar la comparación de los datos se la conoce como “*void compara*”, donde se analizan los datos que se obtienen para saber qué acción tomar ante ellos, como se observa en la **Figura 3.8** se tiene la app-ubicación y la app-bloqueo la cual sirven como determinantes para acciones programadas, dentro de la app-ubicación se obtendrá los datos del GPS y se enviará el mensaje de texto y la app-bloqueo funciona como un bloqueo para el buzzer.

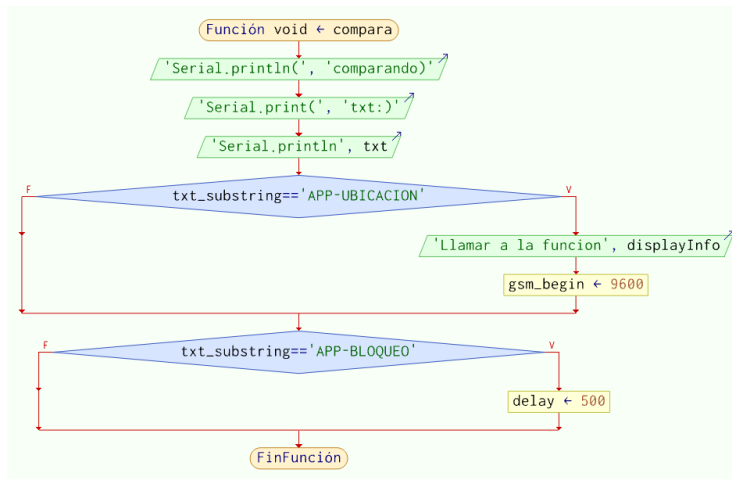


Figura 3.8. Void Compara

La última función es “void sendm1”, la cual se encarga de enviar el mensaje de texto al número previamente agregado, posee una sentencia “Si” la cual nos dirá que al enviarse el mensaje se mostrará en la pantalla serial “enviando” y el texto del mensaje”, si no se envía el mensaje se saldrá de la función sin cumplir otra acción, como se observa en la **Figura 3.9** se tiene el diagrama de la función para el envío del mensaje.

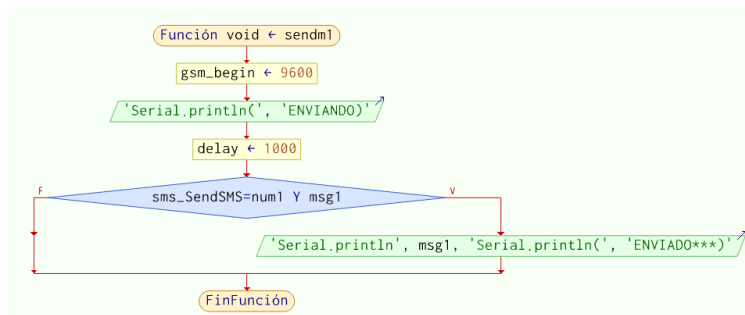


Figura 3.9. Void Sendm1

Armado del circuito del prototipo de velocímetro

Una vez finalizado el código para el prototipo, se realiza el montaje del circuito para comprobar si existe algún error dentro del código y si operan los elementos de manera correcta, como se observa en la **Figura 3.10** se tiene el circuito armado dentro de una “protoboard”, allí se puede observar el respectivo funcionamiento del código dando la señal de velocidad del prototipo, esto quiere decir que cada elemento colocado opera de manera correcta y eficiente dentro del circuito armado del prototipo, otro punto a detallar es que la pantalla muestra el valor de 0 km/h, esto se debe a una prueba inicial del funcionamiento del prototipo estando en un estado de reposo o estático.

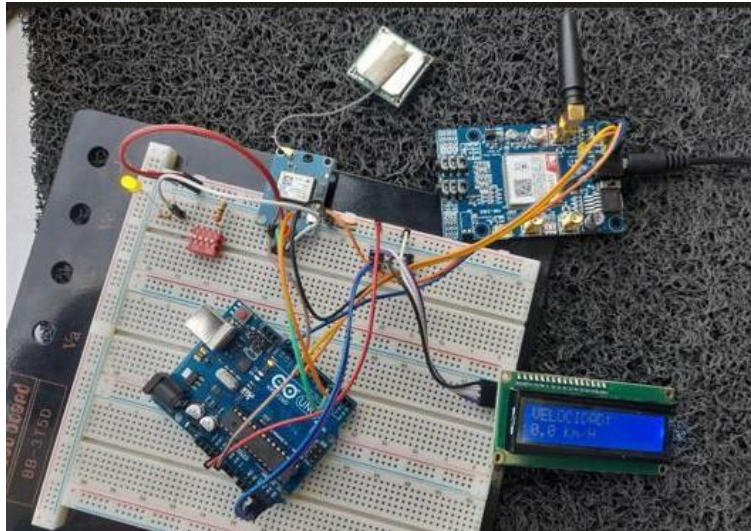


Figura 3.10. Armado Circuito

3.3 Implementación del prototipo del velocímetro

Cuando se obtiene un funcionamiento correcto dentro del circuito armado, se realiza la creación y montaje de la PCB para obtener los datos del módulo GPS a mayores velocidades y obtener una mejor movilidad, a su vez se implementa una cubierta para brindar estabilidad, seguridad al prototipo y se realiza el proceso de instalación dentro de un vehículo para la obtención de los respectivos datos dentro de una carretera a ciertas velocidades.

Creación PCB para el prototipo de velocímetro

Para la creación de una placa de circuito impreso se procede a crear el respectivo circuito dentro del programa llamado "Proteus", para ello se buscan los respectivos elementos que se encuentran registrados en el prototipo, es decir, se buscan los elementos que se encuentran en el armado del prototipo, al encontrar los elementos se colocan en la pantalla del programa y se procede a conectarlos como en el circuito armado, al finalizar con la conexión se procede a realizar el respectivo circuito impreso dentro del programa, dentro de la **Figura 3.11** se presenta el diagrama dentro del programa Proteus.

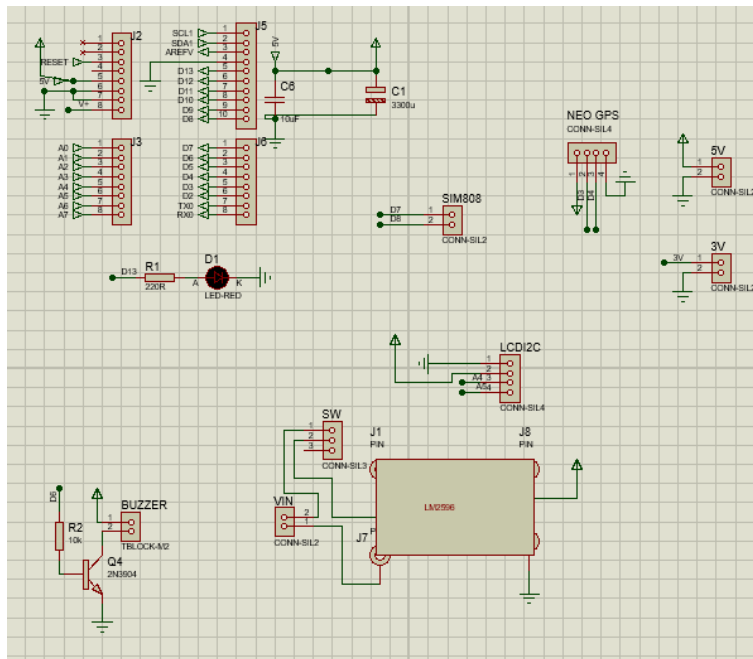


Figura 3.11. Conexión Proteus

Como primer punto se debe crear el espacio de trabajo, es decir, se crea el tamaño de la PCB a armar, al definir el área de trabajo se prosigue a colocar los elementos, al ir colocando los elementos aparecerán las guías de las pistas a colocar.

Todos los elementos que están dentro del área de trabajo para integrarse con las pistas, para ello se dirige a la opción “*track mode*” donde servirá para dibujar las pistas siguiendo las guías de las mismas, otra forma de hacerlo es colocando en la opción “*auto-router*” que designara pistas de forma automática dependiendo de la ubicación de los elementos, como se puede ver en la **Figura 3.12** se tiene el respectivo circuito con las pistas correspondientes para la impresión y colocación en la placa, como punto final se importa a PDF para imprimirlo y colocarlo en la respectiva placa, como se observa en la **Figura 3.13** se tiene un esquema de como irán los elementos conectados en la baquelita.

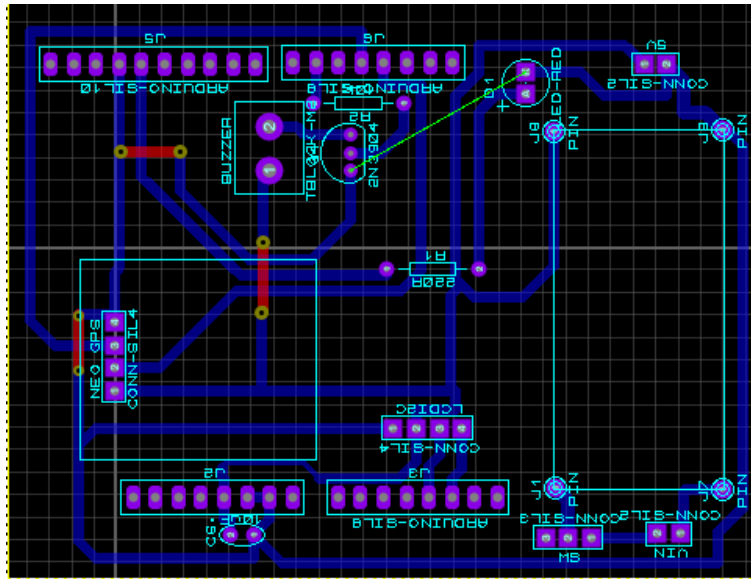


Figura 3.12. Circuito Digital para PCB

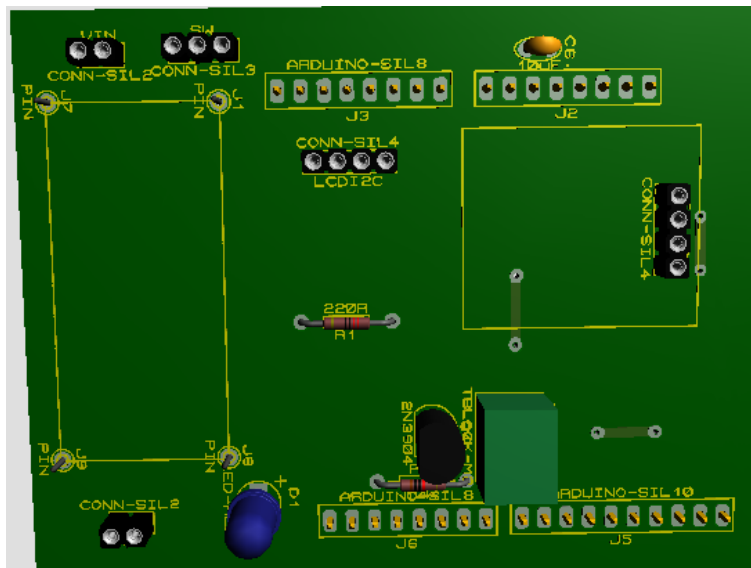


Figura 3.13. Esquema Digital de PCB

Al tener el circuito impreso se procede a pegar dentro de la placa conocida como baquelita, el proceso de transferencia de las pistas del circuito se lo realiza con una plancha para darle el calor, como se puede ver en la **Figura 3.14** se tiene el respectivo planchado del circuito impreso en la baquelita, con el fin de obtener las respectivas pistas pegadas en la placa, para que se queden colocadas se sumerge la placa con el circuito impreso en cloruro férrico, después de 15 minutos se tiene listo el circuito impreso dentro de la baquelita, esto se evidencia en la **Figura 3.15** donde se muestra la colocación de la baquelita en el ácido.



Figura 3.14. Planchado del Circuito en la Baquelita



Figura 3.15. Colocación de Baquelita en Ácido

Cuando se termina la impresión y la limpieza del circuito, se realiza los respectivos orificios para que entren los pines de los elementos, esto se evidencia en la **Figura 3.16** donde se tiene la perforación de la placa con ayuda de una broca para colocar los componentes dentro de la baquelita, una vez hecho este paso se procede a colocar los elementos en su respectiva posición, como se observa en la **Figura 3.17** se presentan los componentes necesarios dentro de la PCB.

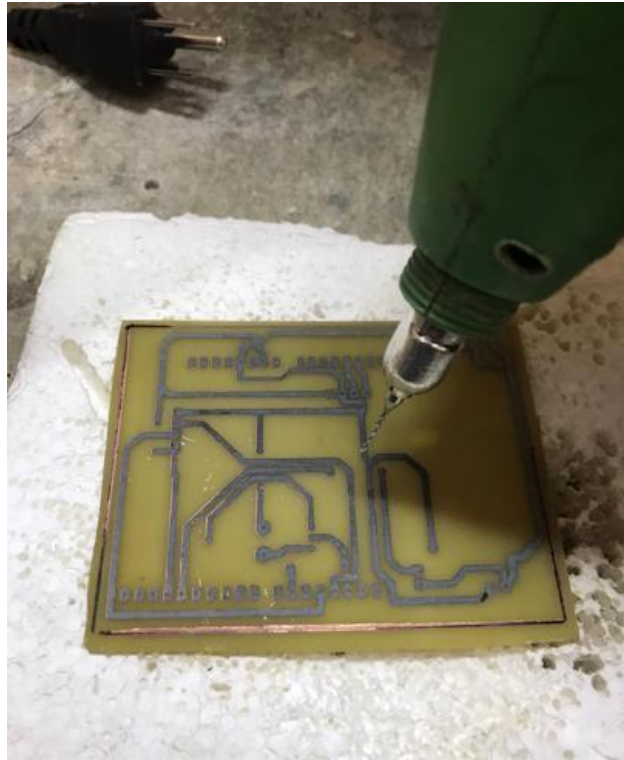


Figura 3.16. Perforación de PCB

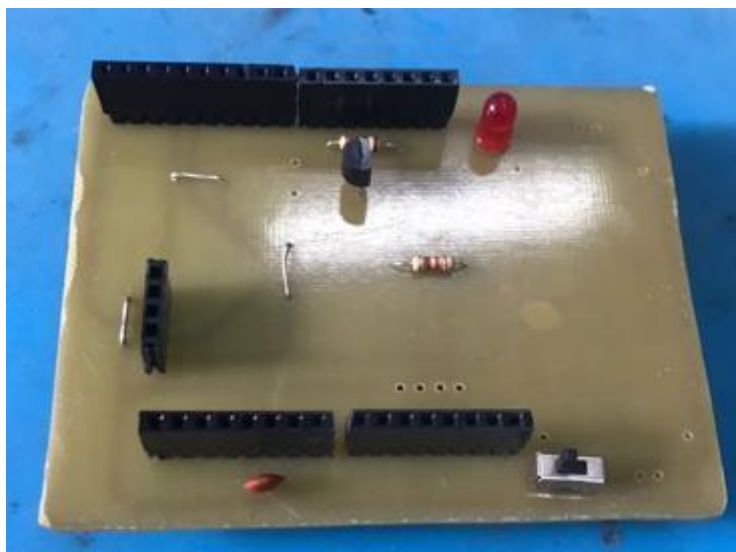


Figura 3.17. Colocación Componentes

Finalmente, se procede a soldar los pines de los componentes dentro de la baquelita, para obtener una comunicación estable entre elementos dentro del circuito, y obtener una mejor estabilidad de los elementos, en otras palabras, obtener una comunicación estable sin que ocurra una desconexión de los mismos con respecto a sus pistas, como se observa en la **Figura 3.18** donde se realiza el proceso de soldadura de los pines de los elementos.

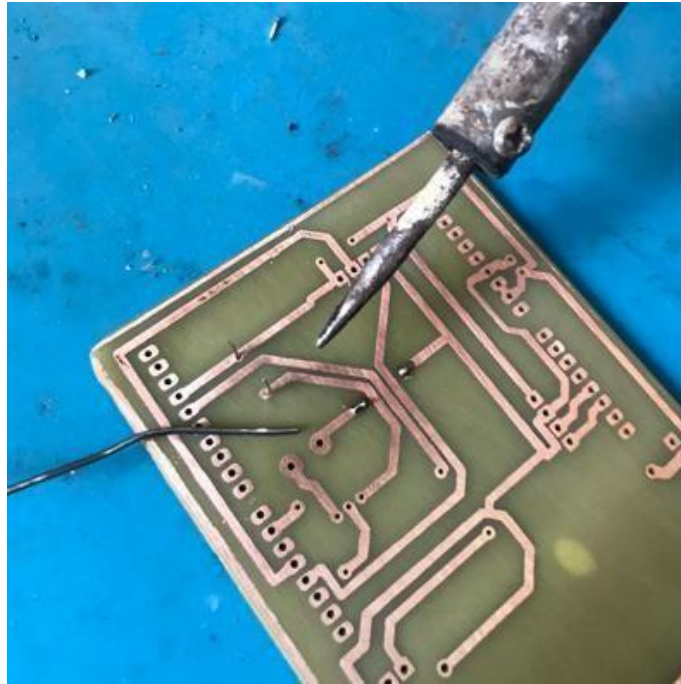


Figura 3.18. Soldadura Componentes

Implementación de cubierta para el prototipo de velocímetro

Para implementar la respectiva cubierta que servirá como protección para el prototipo de velocímetro se procede a realizar mediciones de las respectivas placas y elementos para determinar la medida que tendrá la cubierta, con el fin de obtener una cubierta que se adapte al prototipo se toma una medida con un tamaño con cierto grado mayor a las medidas obtenidas, esto se lleva a cabo para tener espacio libre entre elementos, a su vez se obtiene un circuito con una visibilidad más estética, es decir, no se observara las conexiones y cables que pueden estar sueltos, solo se presentara los componentes del prototipo, como se puede ver en la **Figura 3.19** está la caja de protección y en la **Figura 3.20** se logra ver el prototipo del proyecto.



Figura 3.19. Finalización de Cubierta



Figura 3.20. Colocación de Elementos en la Caja

Instalación del prototipo de velocímetro en un vehículo

Dentro del vehículo a instalar el prototipo de velocímetro, se procede a obtener el respectivo circuito del prototipo de velocímetro con su cubierta ya terminada, para instalar dentro del vehículo en la parte inferior de la cubierta se coloca silicona líquida con ayuda de la respectiva pistola para una correcta colocación, adicional se coloca silicona en el tablero del automóvil, esta acción tiene como propósito obtener una mayor fijación del prototipo dentro del vehículo, como se observa en la **Figura 3.21** se tiene la instalación dentro del vehículo.



Figura 3.21. Instalación Prototipo en Vehículo

3.4 Realizar pruebas de funcionamiento del prototipo

Una vez terminada la instalación del prototipo dentro del vehículo se procede a tomar medidas a mayores velocidades, con el fin de obtener pruebas del funcionamiento y conocer si existe algún tipo de fallo dentro del prototipo de velocímetro, como puede ser al momento de la instalación.

Para dar corriente al circuito se enciende el switch que abre la alimentación para el circuito de todos los elementos del mismo, adicional se activa el módulo GSM para dar paso al envío de los mensajes, como se observa en la **Figura 3.22** se tienen los dos switches que se deben activar para dar una alimentación correcta del prototipo de velocímetro.

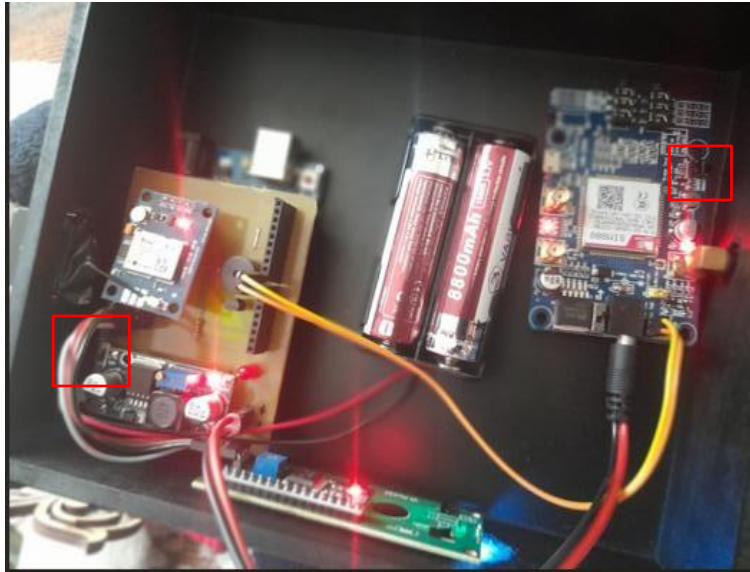


Figura 3.22. Activación de Switches del Prototipo

Al tener el circuito con la alimentación se debe esperar al módulo GPS que reciba datos de los satélites correspondientes, una vez obtenidos los datos de los satélites en el módulo GPS se mostrarán los valores de velocidad en la pantalla LCD y en una computadora los respectivos datos de posición tanto de latitud como longitud que serán almacenados para posteriormente enviados por mensaje cuando ocurra la sentencia de un sobrepaso de velocidad, como se puede ver en la **Figura 3.23** se tiene el módulo GPS antes de recibir los datos de los satélites, en la **Figura 3.24** se tiene los respectivos datos obtenidos de latitud y longitud que se obtienen del módulo GPS mostrados en el monitor serial.

```
15:18:42.221 -> INICIANDO
15:18:59.305 ->
15:18:59.305 -> status=READY
15:18:59.526 -> Location: INVALID
15:18:59.526 -> Location: INVALID
```

Figura 3.23. Datos de Monitor Serial de Módulo GPS al iniciar

```
11:38:26.155 ->
11:38:26.155 -> Location: Speed:0.28
11:38:26.155 -> http://maps.google.com/?q=-0.243419,-78.532951
11:38:26.227 -> http://maps.google.com/?q=-0.243419,-78.532951
```

Figura 3.24. Datos de Monitor Serial de Módulo GPS con datos procesados

Una vez obtenido los datos del módulo GPS se procede a recorrer distancias dentro del vehículo con el prototipo instalado, como se puede ver en la **Figura 3.25** se muestra la pantalla LCD con valores de velocidad tomados dentro de la carretera, y en la **Figura 3.26** se tiene el mensaje de alerta que se obtiene.



Figura 3.25. Pantalla LCD con valores de velocidad

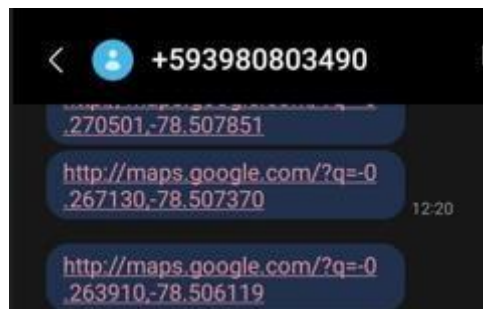


Figura 3.26. Notificación de mensaje en celular del dueño

4. CONCLUSIONES

- Para el siguiente prototipo se instaló un sistema de por medio de pilas de litio, a su vez se implementó un convertidor de 7 voltios a 5 voltios para no causar daños con los elementos del prototipo y como punto final se añadió el sistema de carga para las baterías mediante el cargador USB, con el fin de que el prototipo cumpla con el objetivo propuesto de tener una fuente de alimentación propia y recargable.
- En el presente trabajo se integró varios acoplamientos a la placa Arduino como el GSM, GPS, LCD, entre otros, logrando evidenciar un funcionamiento aceptable del prototipo, donde el aporte del módulo GSM logro la comunicación por medio de mensajes de texto, el GPS incorporo las coordenadas del vehículo en tiempo real integrado en el mensaje de texto y por último el módulo LCD proporciono la información necesaria al usuario sobre su velocidad a la que estaba circulando.
- Durante el desarrollo del prototipo surgieron errores en la compilación del código fuente como también en los elementos utilizados, estos fallos se debieron a que no se usaban las librerías correspondientes para que pueda ejecutarse los comandos dentro del script, adicional el módulo GSM no estaba operando en su totalidad, debido a que el voltaje suministrado tanto para la placa Arduino, el módulo GPS y el módulo GSM no era el necesario para operar.
- La cubierta del prototipo posee características de protección y seguridad para los elementos, cuenta con orificios para que las antenas puedan obtener una señal clara y precisa, otro orificio para la pantalla con el fin de obtener una visualización de los datos que se presentaran, para el retiro de las baterías de litio y la carga de las mismas, la cubierta cuenta con una tapa para abrir y cerrar, otro punto a favor es la accesibilidad hacia los elementos tanto de alimentación como el módulo SIM808, así como el pulsador de reinicio del circuito del módulo Arduino.
- Para las pruebas de funcionamiento se instaló el prototipo de velocímetro dentro de un vehículo, por medio de este medio de transporte se logró verificar varios aspectos, primero comprobar si la fuente de alimentación que cuenta el prototipo funciona correctamente con el puerto USB integrado en el vehículo, por último, se hizo pruebas con las diferentes velocidades tanto bajas, medias y altas, para

determinar el funcionamiento preciso del prototipo, obteniendo los resultados esperados como es el mensaje de texto y la activación del buzzer si la velocidad sobrepasaba el rango establecido.

- Con ayuda del software Proteus se facilitó la conexión de los respectivos elementos del prototipo en un diagrama digital, así como la facilidad de creación del circuito impreso para la placa PCB en baquelita, además de poseer herramientas que permiten el enrutamiento automático de las pistas que contendrá el circuito y la guía para el seguimiento de dónde irán cada una de las pistas en cada elemento.
- El prototipo de velocímetro posee un sistema de protección ante cortocircuitos, el cual se activa produciendo un reinicio del circuito del prototipo y guardando los datos que se tenía antes del cortocircuito dentro de la memoria del microcontrolador, en el proceso de reinicio de los circuitos si algún elemento se descompuso, el código del velocímetro no permitirá el avance a las siguientes etapas.

5. RECOMENDACIONES

- El módulo GPS Neo7M posee un retardo a la hora de conectarse a los satélites, esto se debe al tiempo que tarda en recibir los datos de los satélites y el tiempo que se tarda en procesar los mismos para posteriormente ser procesados por el microcontrolador para la transmisión hacia los elementos faltantes, se recomienda para futuros trabajos se mejore el tiempo de obtención de datos por medio del módulo GPS, mediante la mejora de la calidad de la señal que se obtiene de los satélites.
- La señal que procesa los módulos GPS Neo7M puede ser perdida por construcciones o por falta de reconocimiento de los datos hacia el módulo, cuando ocurra la perdida de la señal del módulo GPS se debe esperar hasta que se procesen los datos y se transmitan.
- Cuando se activa el prototipo se deben activar los dos módulos que son el microcontrolador Arduino y modulo GSM, debido a que si no se activa el primero ningún elemento operará, en cambio si no se activa el GSM no se procederá al envío de mensajes.
- Las baterías de Litio pueden durar días de operación con una carga completa sin necesidad de cargarlas, pero es necesario realizar el proceso de carga de las baterías cuando se presenta a la pantalla LCD con una intensidad baja a la hora de proyectarse datos.

6. REFERENCIAS BIBLIOGRÁFICAS

Referencias

- [1] Arduino.cl, «¿Qué es Arduino?,» 03 enero 2011. [En línea]. Available: <https://arduino.cl/que-es-arduino/>. [Último acceso: 19 febrero 2024].
- [2] BenQ, «LCD ¿Qué es? y ¿Cuál es su uso en monitores?,» 01 abril 2021. [En línea]. Available: <https://www.benq.com/es-mx/centro-de-conocimiento/conocimiento/que-es-lcd-y-como-se-usa-en-monitores.html>. [Último acceso: 19 febrero 2024].
- [3] TuTelefonia.com, «GSM: qué es, para qué sirve y cómo funciona,» [En línea]. Available: <https://www.tutelefonía.com/gsm/>. [Último acceso: 19 febrero 2024].
- [4] Equipo de Expertos en Ciencia y Tecnología, «GSM: definición y usos en las comunicaciones móviles,» [En línea]. Available: <https://www.universidadviu.com/es/actualidad/nuestros-expertos/que-es-gsm-y-como-funciona>. [Último acceso: 19 febrero 2024].
- [5] Electropreguntas, «¿Qué Es Un Buzzer Y Cómo Funciona?,» [En línea]. Available: <https://electropreguntas.com/que-es-un-buzzer-y-como-funciona/>. [Último acceso: 19 febrero 2024].
- [6] S. A. d. L. Sakinah, «Perancangan Simulator EKG (Elektronik Kardiogra Menggunakan Software Proteus 8.0,» septiembre 2017. [En línea]. Available: <https://ejournal.jak-stik.ac.id/index.php/komputasi/article/view/122>. [Último acceso: 19 febrero 2024].
- [7] Elosciloscopio, «Comparación de Arduino vs ESP8266 vs ESP32,» [En línea]. Available: <https://elosciloscopio.com/comparacion-arduino-vs-esp8266-vs-esp32/>. [Último acceso: 19 febrero 2024].
- [8] F. P. Quevedo, «SENSORIZACIÓN INTELIGENTE Y APP MÓVIL PARA BICICLETAS,» junio 2018. [En línea]. Available: <https://crea.ujaen.es/bitstream/10953.1/14314/1/TFG%20-%20Fernando%20Plaza%20Quevedo.pdf>. [Último acceso: 19 febrero 2024].
- [9] Y. Fernández, «Qué es Arduino cómo funciona y qué puedes hacer con uno,» 23 septiembre 2022. [En línea]. Available: <https://www.xataka.com/basics/que-arduino-como-funciona-quepuedes-hacer-uno>. [Último acceso: 19 febrero 2024].
- [10] S. A. G. PAILLACHO, «IMPLEMENTACIÓN DE UN PROTOTIPO PORTÁTIL QUE PERMITA DESPLEGAR SMS PARA PERSONAS DE LA TERCERA EDAD,» marzo 2023. [En línea]. Available: <https://bibdigital.epn.edu.ec/bitstream/15000/23781/1/CD%2013047.pdf>. [Último acceso: 19 febrero 2024].
- [11] M. Z. J. Hidalgo, «IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO ELECTRÓNICO PARA LA ADMINISTRACIÓN DE UN LOCUTORIO TELEFÓNICO,» 2007. [En línea]. Available: <http://dspace.espe.edu.ec/bitstream/123456789/1090/1/38T00104.pdf>. [Último acceso: 19 febrero 2024].
- [12] V. Quintero, «Baterías de ion litio: características y aplicaciones,» 08 enero 2021. [En línea]. Available: <https://revistas.utp.ac.pa/index.php/id-tecnologico/article/view/2907/3616>. [Último acceso: 19 febrero 2024].

7. ANEXOS

ANEXO I: Certificado de Originalidad

Quito, D.M. 21 de febrero de 2022

De mi consideración:

Yo, CARLOS ANDRES YUNGA SANCHEZ, en calidad de directora del Trabajo de Integración Curricular titulado IMPLEMENTACIÓN DE UN PROTOTIPO DE VELOCÍMETRO QUE NOTIFIQUE EXCESOS DE VELOCIDAD VÍA SMS asociado al IMPLEMENTACIÓN DE UN PROTOTIPO DE VELOCÍMETRO QUE NOTIFIQUE EXCESOS DE VELOCIDAD VÍA SMS elaborado por el estudiante JUAN CAARLOS VALDEZ BETANCOURT de la carrera en Tecnología Superior en Redes y Telecomunicaciones, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 10 %.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el link del informe generado por la herramienta Turnitin. LINK

[3. Reporte de Turnitin Juan Valdez Tic 2023B.pdf](#)

Atentamente,

CARLOS ANDRÉS YUNGA SÁNCHEZ

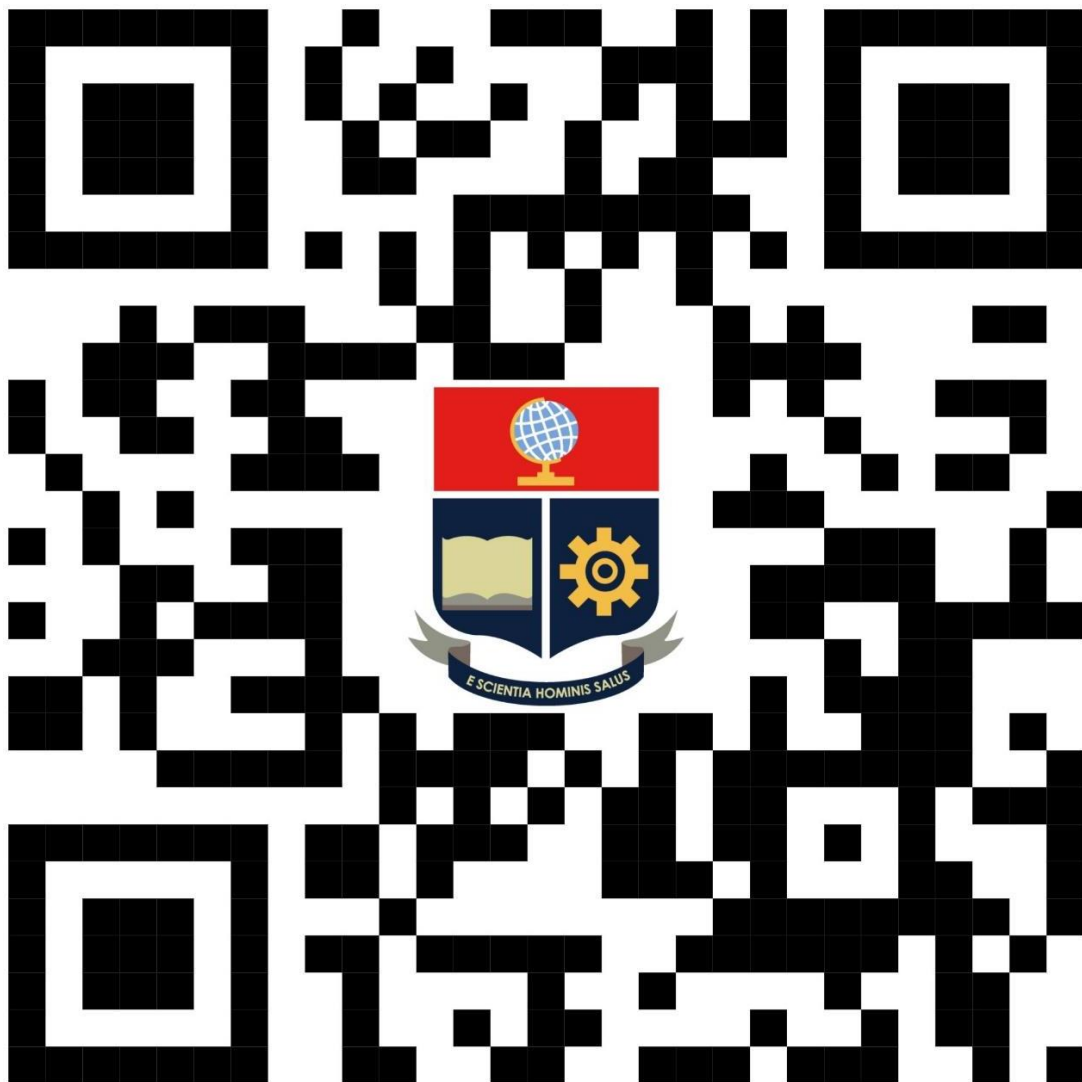
Técnico Docente

Escuela de Formación de Tecnólogos

ANEXO II: Enlaces

A continuación, se presenta el enlace del video de demostrativo del prototipo de velocímetro con su respectivo funcionamiento a las diferentes velocidades en la carretera y el código QR del video:

<https://youtu.be/OfIAfEiVXgc>



ANEXO III: Código Fuente

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#include <sms.h>
SMSGSM sms;
TinyGPSPlus gps;
SoftwareSerial ss(4, 3);
char telefono[13]="0991467259"; // Número principal
char mensaje1[50]="NO HAY SENAL GPS";
String txt;
float vl;
boolean started=false;
char mensaje_ubicacion=0;
char numero_telefono[20];
char texto_mensaje[100];
int i;
String txt3;
int zumbador=6;

void setup() {
  pinMode(zumbador,OUTPUT);
  lcd.init();
  lcd.backlight();
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("INICIANDO");
  digitalWrite(zumbador,HIGH);
  delay(200);
  digitalWrite(zumbador,LOW);
  Serial.begin(9600);
  Serial.println("INICIANDO");
  ss.begin(9600);
  if (gsm.begin(9600))
  {
    Serial.println("\nstatus=READY");
    started=true;
  }
  else
    Serial.println("\nstatus=IDLE");
  if(started)
  {int i;
  }
}
```



```

void loop() {
for(int x=1;x<10;x++){
if(v1>90.0){
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("ENVIANDO SMS");
  sendm1();x=20;
  delay(5000);
  digitalWrite(zumbador,HIGH);
}else{
  digitalWrite(zumbador,LOW);
}
}
delay(1000);
leergps();
}
}

void leergps(){
  ss.begin(9600); // Reinicio del GPS
  delay(100);
  while (ss.available() > 0)
    if (gps.encode(ss.read()))
      displayInfo();
  if (millis() > 5000 && gps.charsProcessed() < 10)
  {
    Serial.println(F("No GPS detected: check wiring."));
  }
}

void displayInfo()
{
  Serial.print(F("Location: "));
  if (gps.location.isValid())
  {
    float latitud, longitud;
    float velocidad1;
    latitud=gps.location.lat();
    longitud=gps.location.lng();
    char enlace[50]="http://maps.google.com/?q="; //arma la trama para envi 
SMS con link directo maps
    char lat [15];
    char lon [15];
    dtostrf(latitud,3,6,lat); //transforma de float a car cter
    dtostrf(longitud,3,6,lon);
    velocidad1=gps.speed.kmph();
    v1=velocidad1;
    Serial.print("Speed:");
    Serial.println(velocidad1);

```

```

    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("VELOCIDAD:");
    lcd.setCursor(0,1);
    lcd.print(velocidad1,1);
    lcd.print(" Km/H");
    char trans [10];
    int t=analogRead(A1)/2;
    sprintf (trans, "%s", lat);
    strcat(enlace,trans);
    sprintf (trans, ",%s", lon);
    strcat(enlace,trans);
    Serial.println(enlace);
    for (int x=0;x<50;x++){
    mensaje1[x]=enlace[x];
    }
    Serial.println(mensaje1);
    }
    else
    {
        Serial.print(F("INVALID"));
    }
    Serial.println();
}

void compara(){
    Serial.println("comparando");
    Serial.print("txt:"); Serial.println(txt);
    if(txt.substring(0)=="APP-UBICACION"){ // el texto recibido es "app-
    ubicación"
        displayInfo(); //obtiene la posición del GPS
        gsm.begin(9600);
        sendm1(); //ENVIA EL MENSAJE ESCRITO
        }
    if(txt.substring(0)=="APP-BLOQUEO"){
        delay(500);
        }
    }

void recunread(){
    gsm.begin(9600);
    txt="";
    Serial.println("recunread"); //lea el buzón de mensaje nuevos y devuelva la
    posición del mismo si es 0 no hay mensajes nuevos
    mensaje_ubicacion=sms.IsSMSPresent(SMS_UNREAD); //si es diferente de 0 lea el
    SMS
        if (mensaje_ubicacion)
        {
            Serial.print("SMS postion:");

```

```

        Serial.println(mensaje_ubicacion,DEC);
        sms.GetSMS(mensaje_ubicacion, numero_telefono, texto_mensaje, 100)
        //lee el SMS y devuelve posición, número y contenido del mensaje
        Serial.println(numero_telefono);
        int x;

        txt="";
        for(x=0;x<20;x++){
            txt=txt+texto_mensaje[x]; //guardo el contenido del mensaje en la variable
            txt, carácter por carácter
        }
        txt=String(texto_mensaje);
        txt.trim();
            Serial.println(texto_mensaje);
            Serial.println(txt);
            compara (); //función compara el contenido del SMS
        }
    delay(2000);
}

void sendm1(){
    gsm.begin(9600);
    Serial.println("ENVIANDO");
    delay(1000);
    if (sms.SendSMS(telefono, mensaje1)) //envíe un SMS al telefono con el
    contenido de la variable mensaje1
    {
        Serial.println(mensaje1);Serial.println("ENVIADO***");
    }
}

```