



“Responsabilidad con pensamiento positivo”

UNIVERSIDAD TECNOLÓGICA ISRAEL

**TRABAJO DE TITULACIÓN EN OPCIÓN AL GRADO DE:
INGENIERO EN ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES**

TEMA:

**DESARROLLAR UN DISPOSITIVO ELECTRÓNICO QUE SIRVA COMO CAJA NEGRA
PARA ALMACENAMIENTO DE INFORMACIÓN PARA VEHÍCULOS**

AUTOR:

CEDEÑO ANZULES LIZARDO DANIEL

TUTOR:

.....
Mg. Francisco Jurado

QUITO, ECUADOR

2018

UNIVERSIDAD TECNOLÓGICA ISRAEL**APROBACIÓN DEL TUTOR**

En mi calidad de tutor del trabajo de titulación certifico:

Que el trabajo de titulación “**DESARROLLAR UN DISPOSITIVO ELECTRÓNICO QUE SIRVA COMO CAJA NEGRA PARA ALMACENAMIENTO DE INFORMACIÓN PARA VEHÍCULOS**”, presentado por el Sr. Lizardo Daniel Cedeño Anzules, estudiante de la Carrera de Electrónica y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito D. M., Agosto de 2018

TUTOR

Mg. Francisco Jurado

AGRADECIMIENTO

En primer lugar, agradezco a mis padres quienes fueron un pilar fundamental en mi vida diaria que siempre me impulsaron a luchar por mis metas en donde me encuentre.

También de manera muy especial a mi hijo y sobre todas las cosas a mi esposa, que siempre me acompaña en las circunstancias de la vida para superar todos los obstáculos que cada día fortalecen nuestros lazos por el bienestar de nuestra familia.

DEDICATORIA

Dedico cada instante de mis esfuerzos en la culminación de una etapa de mi vida universitaria a mi esposa y familia que siempre fueron un aliento de mejora en todas las circunstancias que se presentaron a lo largo de mi carrera.

Como también a mis buenos amigos que coseche en una larga lucha de adversidades que reforzaban nuestros conocimientos al superarlos de manera eficiente y con el agrado de conocer a profesionales que impulsaban un símbolo de mejora profesional gracias a sus grandes consejos y experiencias.

TABLA DE CONTENIDO

AGRADECIMIENTO	iii
DEDICATORIA.....	iv
RESUMEN.....	xi
ABSTRACT	xii
INTRODUCCIÓN.....	1
OBJETIVO GENERAL	2
OBJETIVOS ESPECÍFICOS	2
CAPÍTULO I.....	4
1. FUNDAMENTACIÓN TEÓRICA	4
1.1. MICROPROCESADORES	5
1.2. MICROCONTROLADORES	6
1.2.1. MICROCONTROLADOR ATMEGA328	6
1.3. SENSORES	7
1.3.1. FUNCIONAMIENTO.....	8
1.3.2. SEÑALES	8
1.3.3. NATURALEZA DE FUNCIONAMIENTO	8
1.3.4. ELEMENTOS DE FABRICACIÓN	9
1.4. SENSORES DE AGUA	10
1.5. SENSORES DE GAS.....	11
1.6. SENSOR ACELERÓMETRO- GIROSCOPIO	11
1.7. SISTEMA GSM	13
1.8. SISTEMA GPS.....	14
1.9. MINICOMPUTADORAS	16
2. PROPUESTA	18
2.1. MÓDULOS DEL PROYECTO.....	20
2.1.1. ETAPA DE ALIMENTACIÓN.....	21
2.1.2. ETAPA DE CONTROL	22

2.1.3. ETAPA DE ACTUADORES	23
2.1.3.1. ACELERÓMETRO/ GIROSCÓPIO	23
2.1.4. ETAPA DE SENSORES	24
2.1.4.1. SENSOR DE AGUA.....	25
2.1.4.2. SENSOR DE HUMO.....	25
2.1.5. ETAPA DE COMUNICACIÓN.....	26
2.1.5.1. GSM SIM 800	26
2.1.5.2. GPS NEO 6M.....	27
2.1.6. ETAPA DE GRABACIÓN.....	28
2.1.6.1. RASPBERRY PI3	28
2.1.6.2. CÁMARA HD.....	29
2.1.6.3. MICRÓFONO.....	30
2.1.7. ETAPA DE ALMACENAMIENTO	30
2.1.7.1. MICRO SD.....	31
2.2. ASPECTOS TÉCNICOS DEL PRODUCTO EN SU TOTALIDAD	31
2.3. ANÁLISIS DE COSTOS.....	32
2.4. TIEMPO REQUERIDO.....	36
CAPÍTULO III	39
3. IMPLEMENTACIÓN	39
3.1. DESARROLLO	39
3.2. DISEÑO.....	40
3.2.1. ALIMENTACIÓN DE LOS MÓDULOS ARDUINO Y RASPBERRY	43
3.2.2. DISEÑO DE LA PLACA PCB PARA LA ADAPTACIÓN DE LOS ELEMENTOS	44
3.2.3. CONEXIÓN DE LOS MÓDULOS DE LA ETAPA DE CONTROL.....	46
3.2.4. CONEXIÓN DE LA ETAPA DE COMUNICACIÓN.....	50
3.2.5. ELABORACIÓN DE LA ETAPA DE ALMACENAMIENTO	51
3.2.6. DISEÑO DE LA CAJA DEL PROYECTO.....	53
3.3. IMPLEMENTACIÓN DEL PROYECTO.....	55
3.3.1. ETAPA DE ALIMENTACIÓN	55
3.3.2. ETAPA DE CONTROL.....	56

3.3.3. ETAPA DE GRABACIÓN	62
3.3.4. ETAPA DE ALMACENAMIENTO	63
3.4. PRUEBAS DE FUNCIONAMIENTO	66
3.4.1 ENCENDIDO DE DISPOSITIVOS	66
3.4.2 ARRANQUE DE LOS SENSORES.....	68
3.4.3 ALMACENAMIENTO DE INFORMACIÓN	69
3.4.4 ENVÍO DE INFORMACIÓN	70
3.4.5 PRUEBAS DE EVENTOS:	72
3.5. ANÁLISIS DE RESULTADOS	74
CONCLUSIONES.....	77
RECOMENDACIONES	78
REFERENCIAS BIBLIOGRAFICAS	79
ANEXOS	80
1. ANEXO CONFIGURACIÓN ARDUINO.....	81
2. ANEXO PROGRAMACIÓN GPS	82
3. ANEXO 2 INTREGRACION ARDUINO.....	90
4. ANEXO CONFIGURACION RASPBERRY.....	99
5. ANEXO CONFIGURACION DE COMUNICACION ARDUINO.....	102

LISTA DE FIGURAS

Figura. 1. 1 Funcionamiento de Microprocesadores	5
Figura. 1. 2 Funcionamiento de un Microcontrolador.....	6
Figura. 1. 3 Microcontrolador Atmega 328p.....	7
Figura. 1. 4 Clasificación según su funcionamiento.....	8
Figura. 1. 5 Clasificación según sus señales.....	8
Figura. 1. 6 Clasificación según su funcionamiento.....	9
Figura. 1. 7 Clasificación según su fabricación.....	10
Figura. 1. 8 Sensor de Agua	10
Figura. 1. 9 Sensor MQ3	11
Figura. 1. 10 Funcionamiento Sensor Acelerómetro.....	12
Figura. 1. 11 Giroscopio y sus ejes.....	12
Figura. 1. 12 Arquitectura del sistema GSM	14
Figura. 1. 13 Sistema GPS.....	16
Figura. 1. 14 Minicomputador Raspberry	17
Figura. 2. 1 Diagrama Lógico de la Caja Negra.....	19
Figura. 2. 2 Secuencia de la Caja Negra.....	20
Figura. 2. 3 Módulos de funcionamiento.....	21
Figura. 2. 4 Regulador convertidor DC-DC	22
Figura. 2. 5 Arduino Mega	22
Figura. 2. 6 Acelerómetro- Giroscopio.....	24
Figura. 2. 7 Sensor de Agua	25
Figura. 2. 8 Sensor de Humo	26
Figura. 2. 9 Conversor USB Serial.....	26
Figura. 2. 10 Módulo GSM SIM 800	27
Figura. 2. 11 GPS NEO 6M.....	27
Figura. 2. 12 Conexión de una Raspberry Pi3.....	28
Figura. 2. 13 Raspberry y Conversor USB Serial.....	29
Figura. 2. 14 Cámara HD Genius	29
Figura. 2. 15 Micrófono.....	30
Figura. 2. 16 Adaptador de Lectura Micro SD.....	31
Figura. 2. 17 Diagrama de GANTT.....	38
Figura. 3. 1 Diagrama de Bloques de la Caja Negra	39

Figura. 3. 2 Diseño del dispositivo electrónico para almacenamiento de información caja negra	42
Figura. 3. 3 Diagrama de conexión de conversor DC-DC a 5V	44
Figura. 3. 4 Diseño de placa PCB.....	45
Figura. 3. 5 Diagrama de conexión del Sensor de Agua	46
Figura. 3. 6 Diagrama de conexión del Sensor de Agua	47
Figura. 3. 7 Diagrama de conexión del Módulo GSM SIM 800	48
Figura. 3. 8 Diagrama de conexión del Módulo GPS NEO 6M	49
Figura. 3. 9 Diagrama de conexión del Módulo Acelerómetro-Giroscopio	49
Figura. 3. 10 Diagrama de conexión del Módulo Arduino Mega y Raspberry Pi	50
Figura. 3. 11 Diagrama de conexión de la Raspberry Pi con la Cámara y Micrófono	51
Figura. 3. 12 Diagrama de conexión de la tarjeta de almacenamiento del programa de la Raspberry Pi	51
Figura. 3. 13 Diagrama de conexión de la tarjeta de almacenamiento de información.....	52
Figura. 3. 14 Diseño Mecánico de Caja Plástica	54
Figura. 3. 15 Distribución de elementos.....	54
Figura. 3. 16 Diseño Mecánico para la Cámara y GPS	55
Figura. 3. 17 Conversor de voltaje DC-DC	56
Figura. 3. 18 Colocación del conversor de voltaje DC-DC	56
Figura. 3. 19 Colocación del Módulo Arduino Mega.....	57
Figura. 3. 20 Impresión de placa PCB.....	57
Figura. 3. 21 Adaptación de los elementos a la placa PCB	58
Figura. 3. 22 Adaptación de Modulo GSM SIM 800 a la placa PCB	58
Figura. 3. 23 Conexión de Bus de datos de 4 Vías.....	59
Figura. 3. 24 Conexión de Bus de datos para Tx y Rx	59
Figura. 3. 25 Conexión de Acelerómetro-Giroscopio en placa PCB.....	60
Figura. 3. 26 Identificación de pines del Sensor de Humo	61
Figura. 3. 27 Conexión terminales del Sensor de Agua	61
Figura. 3. 28 Identificación de puertos USB de la Raspberry Pi.....	62
Figura. 3. 29 Caja de la Cámara HD y Micrófono	63
Figura. 3. 30 Conexión de tarjeta de almacenamiento del Programa	63
Figura. 3. 31 Conexión de tarjeta de almacenamiento del Programa	64
Figura. 3. 32 Colocación de la Caja en parte inferior del tablero.....	65
Figura. 3. 33 Colocación de la Caja de la cámara en el parabrisas	65
Figura. 3. 34 Colocación del sensor de agua en el guardalado.....	66

LISTA DE TABLAS

Tabla. 2. 1 Tipos de Arduino.....	23
Tabla. 2. 2 Rangos de operación.....	24
Tabla. 2. 3 Almacenamiento de Archivos	31
Tabla. 2. 4 Lista de materiales para construcción del proyecto.....	33
Tabla. 2. 5 Descripción de cotizaciones	34
Tabla. 2. 6 Costo horas Hombre	35
Tabla. 2. 7 Monto de inversión.....	35
Tabla 3. 1 Distribución de cargas Módulo Arduino Mega	43
Tabla 3. 2 Distribución de cargas Módulo Raspberry	43
Tabla 3. 3 Rango de configuración de Sensor de Agua.....	47
Tabla 3. 4 Rango de configuración de Sensor de Humo	47
Tabla 3. 5 Dimensión de Memoria para Raspberry.....	52
Tabla 3. 6 Tamaño de archivos en almacenamiento.....	53
Tabla 3. 7 Checklist de encendido de dispositivos errónea	67
Tabla 3. 8 Checklist de encendido de dispositivos corregidos verificada	68
Tabla 3. 9 Checklist de funcionamiento de los sensores	68
Tabla 3. 10 Checklist de ejecución de los sensores verificada.....	69
Tabla 3. 11 Checklist de ejecución de pruebas de almacenamiento de información errónea ...	69
Tabla 3. 12 Checklist de ejecución de pruebas de precisión del GPS NEO 6M	70
Tabla 3. 13 Checklist de ejecución de pruebas de almacenamiento de información verificada	70
Tabla 3. 14 Checklist de ejecución de pruebas de envío de información errónea.....	71
Tabla 3. 15 Checklist de pruebas envío de información con las operadoras móviles	71
Tabla 3. 16 Checklist de ejecución de pruebas envío de información verificada	72
Tabla 3. 17 Prueba de Cambio Brusco de Velocidad.....	72
Tabla 3. 18 Prueba de cambio brusco de velocidad verificada	73
Tabla 3. 19 Prueba de volcamiento	73
Tabla 3. 20 Pruebas del sensor de Humo.....	74
Tabla 3. 21 Resultado de pruebas del Sensor de Agua.....	74
Tabla 3. 22 Resultados de posición del GPS	75
Tabla 3. 23 Verificación Red Móvil.....	76

RESUMEN

La caja negra para vehículos, es un proyecto que se enfoca en almacenar información específica para determinar la posible causa de un accidente de tránsito provocado por el vehículo.

Este dispositivo tiene la finalidad de monitorear y guardar información sobre la velocidad, ubicación, hora y fecha, del vehículo, gracias al uso de un GPS integrado.

Adicionalmente contiene un acelerómetro-giroscopio que identifica el movimiento del vehículo y determina un evento, entiéndase como evento al choque, volcamiento o incendio en el interior del vehículo.

También integra un sensor de humo para detectar un conato de incendio, además posee un sensor de agua, el cual permite evaluar la cantidad de agua en la calzada y así indicar al conductor para que tome las debidas precauciones de manejo.

Al ocurrir un evento, entra en funcionamiento una cámara y un micrófono, los cuales sirven para grabar y registrar esta información.

La información que corresponde a la ubicación, velocidad, hora y fecha, se guarda en una memoria MICRO SD como un archivo .TXT. Adicionalmente las grabaciones de audio y video, generan dos archivos independientes de aproximadamente un minuto, que se almacenan en la misma memoria.

Las notificaciones de choque, volcamiento y conato de incendio se realizan mediante un mensaje de texto.

PALABRAS CLAVE: Implementación, Investigación, Accidentes, Caja Negra, Vehículos

ABSTRACT

The black box for vehicles, is a project that focuses on specific information to determine the possible cause of an accident caused by the vehicle.

This device has the purpose of monitoring and saving information about the speed, location, time and date of the vehicle, thanks to the use of an integrated GPS.

It also contains an accelerometer-gyroscope that identifies the movement of the vehicle and determines an event, understood as an event to shock, overturn or fire inside the vehicle.

It also integrates a smoke sensor to detect a fire, in addition to a water sensor, which allows to evaluate the amount of water in the road and thus indicate the driver to take the necessary precautions.

When an event occurs, start a camera and a microphone, things that can record and record this information.

The information corresponding to the location, speed, time and date is stored in a MICRO SD memory as a .TXT file. In addition, audio and video recordings generate two independent files of approximately one minute, which are stored in the same memory.

Shock, rollover, and fire congestion notifications are made via a text message.

KEY WORDS: Implementation, Investigation, Accidents, Black Box, Vehicles

INTRODUCCIÓN

Numerosos factores pueden incidir en una colisión vehicular. Entre estos es necesario destacar: negligencia de los peatones al cruzar las vías, inobservancia de las leyes de tránsito por parte de los conductores, ingesta de alcohol, uso del teléfono celular mientras se conduce, avería fortuita del vehículo, etc. En algunos casos determinar las causas de un determinado incidente puede ser sumamente difícil.

Los acontecimientos en un accidente de tránsito en lugares donde no existe vigilancia por cámaras públicas o por un agente de tránsito, son más complicados para determinar cuál fue la causa del percance. De esta manera, las partes involucradas tratan de justificar su acción al volante de manera inocente.

En atención a los casos de accidentes de tránsito, según la OMS (Organización Mundial de la Salud) en el ámbito de la Seguridad Vial propone reducir las cifras previstas de víctimas mortales en AT a través de 5 pilares básicos (Organización Mundial de la Salud, 2015).

1. Gestión de la seguridad vial
2. Vías de tránsito y movilidad
3. Vehículos más seguros
4. Usuarios de vías de tránsito
5. Respuesta tras los accidentes

Actualmente la mitad de las víctimas mortales son peatones, ciclistas y motociclistas, ya que el contacto en un accidente de tránsito es directo y sin ningún mecanismo de reducción de impacto (Organización Mundial de la Salud, 2015).

Con base en los antecedentes expuestos, se diseña el presente proyecto con la finalidad de obtener datos verídicos de un accidente. De esta manera se puede utilizar esta información en temas legales, que servirán como evidencia de los hechos gracias a la información obtenida de la caja negra que se encuentra ubicada al interior del vehículo.

La utilización de un sistema de almacenamiento de audio, video y diferentes velocidades a las que circulan los automotores; son útiles en los casos de justificar la inocencia o culpabilidad de un conductor.

De esta manera se evitará que un conductor sea juzgado por la ley de manera injusta cuando no sea el culpable de un accidente o a su vez a que un conductor sea sancionado con todo el peso de la ley por ser el causante de un accidente de tránsito al tener la evidencia fiable del lugar de los hechos.

OBJETIVO GENERAL

Desarrollar un dispositivo electrónico que sirva como caja negra para almacenamiento de información para vehículos

OBJETIVOS ESPECÍFICOS

- Diseñar un dispositivo electrónico con los requerimientos necesarios que permitan almacenar la información de video y audio en una micro SD.
- Desarrollar un programa que permita integrar la lectura y comunicación, entre los sensores de agua, sensor de humo y acelerómetro - giroscopio.
- Implementar un sistema de alarma que notifique al dueño del auto y un familiar cercano mediante un mensaje de texto un posible riesgo detectado.
- Elaborar una placa de conexión con Arduino Mega2560 y los sensores de agua, humo y acelerómetro-giroscopio, para una mejor interacción.
- Comprobar el funcionamiento del GPS para del sistema de almacenamiento de un archivo.TXT en la memoria micro SD.

DESCRIPCIÓN DE LOS CAPÍTULOS

CAPÍTULO I

En este capítulo se realiza la fundamentación de la idea del proyecto, se basa en las teorías y procedimientos que permiten interactuar con todos los elementos que constituyen el proyecto.

CAPÍTULO II

El presente capítulo refleja el detalle de los elementos que se utilizan para la construcción y elaboración de las funciones del dispositivo, los elementos más importantes que se definen son el Módulo Arduino Mega y la Raspberry Pi3 que se encargan de procesar las señales de los sensores y actuadores que cumplen una función especial en toda la propuesta del proyecto.

CAPÍTULO III

La implementación del proyecto permite realizar la conexión de cada sensor y actuador según su funcionamiento, además de coincidir con los detalles de cada elemento que se verifica en el CAPÍTULO II. Las facilidades de conexión y desconexión de los elementos permiten realizar la configuración de los mismos de manera oportuna que confirma su funcionamiento.

CAPÍTULO I

1. FUNDAMENTACIÓN TEÓRICA

Un dispositivo electrónico es un conjunto de elementos electrónicos organizados en un circuito, que se encuentran listos para cumplir una función específica que controla y aprovecha las señales eléctricas.

A los fines de este proyecto se define como caja negra para vehículos a un dispositivo electrónico, que permitirá a los usuarios almacenar información de lo que sucede en el exterior como en el interior del vehículo en un accidente; que se identifica como choque, volcamiento o conato de incendio en el interior del vehículo.

Existen varias versiones comerciales de cajas negras para vehículos. Entre estas se puede mencionar el Car Ángel BBX1 (SitiosArgentina, 2016). El Car Ángel BBX1 registra video, ubicación (provista por GPS), velocidad, fecha y hora en correspondencia de cada cambio brusco de velocidad y almacena los datos correspondientes en una tarjeta microSD.

El grupo Volkswagen también se ha sumado a la innovación en este campo con el desarrollo de un dispositivo llamado DiBox. El objetivo principal de este aparato es registrar la actividad relacionada con el funcionamiento y el desplazamiento del vehículo, y permite, además, conectarse a un dispositivo móvil y visualizar las actividades en una aplicación del dispositivo (EL COMERCIO, 2015).

La caja negra se puede implementar en distintos medios de movilización, que deseen poseer información detallada en caso de sufrir algún accidente. Así obtendrán información veraz y fundamentada.

Con la implementación de estos dispositivos se pretende mejorar el desarrollo y funcionamiento, para esto, añade un sistema de alarma sonora en caso de existir una señal de humo en el interior del vehículo y una notificación vía SMS (Short Message Service) de los eventos que se presentan en la movilización del dispositivo, así como también un sensor de agua que informa al conductor que se moviliza en una carretera con agua.

Los microcontroladores son fundamentales, ya que se encuentran instalados en la mayoría de dispositivos electrónicos, esto dado por la facilidad de acople con todos los equipos, a su vez, también existen varios tipos de microcontroladores.

1.1. MICROPROCESADORES

Los microprocesadores se han situado en el mercado tecnológico y requieren de una gran capacidad de velocidad, ejecución y memoria. Un microprocesador es un circuito lógico que responde y procesa las operaciones lógicas y aritméticas, que hacen funcionar a las computadoras, en conclusión, es el cerebro de un computador. En la Figura 1.1 se representa esquemáticamente su funcionamiento.

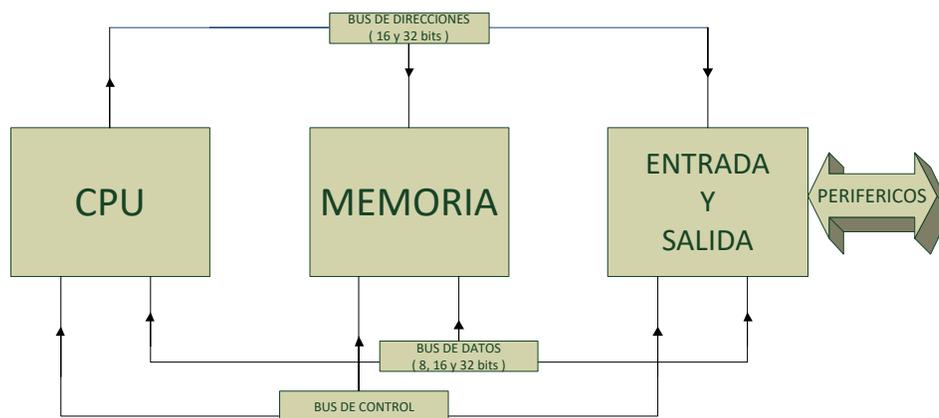
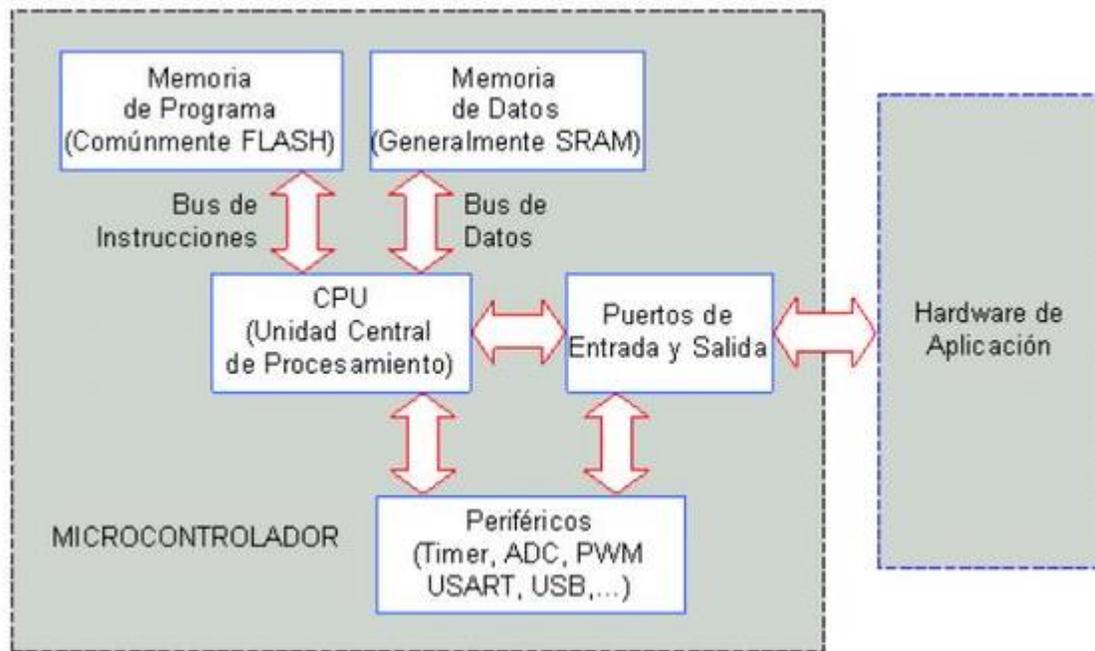


Figura. 1. 1 Funcionamiento de Microprocesadores
Fuente: (Valdés Fernando, 2007)

1.2. MICROCONTROLADORES

Un microcontrolador es un circuito integrado programable, que es capaz de ejecutar órdenes almacenadas en su memoria. Está compuesto de varios bloques funcionales que cumplen una tarea asignada, también incluye en su interior las tres principales unidades de una computadora; que son: unidad central de procesamiento, memoria y periféricos de entrada/salida que se detallan en la Figura 1.2.



*Figura. 1. 2 Funcionamiento de un Microcontrolador
Fuente: (Valdés Fernando, 2007)*

1.2.1. MICROCONTROLADOR ATMEGA328

Uno de los microcontroladores que se encuentran en las plataformas de Arduino es el Atmega328p, como se puede apreciar en la figura 1.3. Este es un microcontrolador que está basado en la arquitectura AVR (Automatic Voltage Regulator) AVR es una arquitectura basada en RISC (Computadora con conjunto de instrucciones reducidas), es importante recordar que la arquitectura de un microprocesador o microcontrolador tiene una estructura

interna que permite la distribución, acceso y control de los diversos bloques que lo componen (Muhammad, 2014).

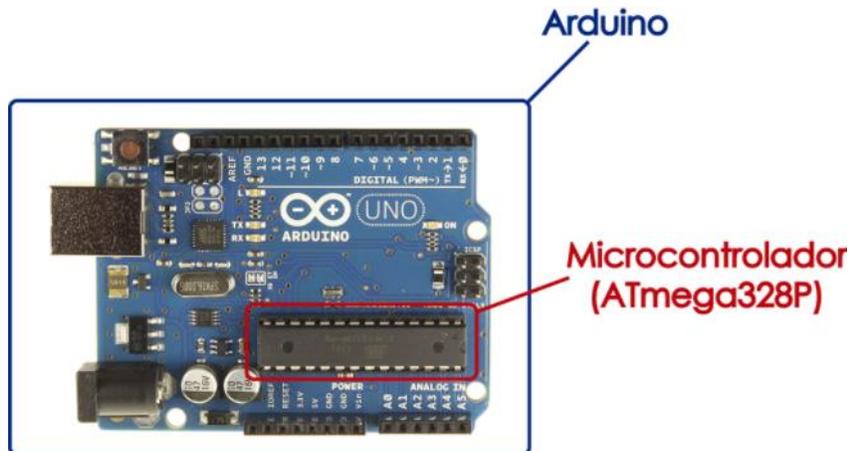


Figura. 1. 3 Microcontrolador Atmega 328p
Fuente: (García, 2105)

Esta tarjeta electrónica cuenta con un microcontrolador ATmega328 que incluye 14 entradas/salidas digitales, 6 entradas analógicas, un cristal cerámico de 16MHz y conexión USB

1.3. SENSORES

Son imitaciones de los sentidos de los seres humanos, tiene una propiedad sensible en magnitud al medio, existen sensores tecnológicos relacionados con diferentes sentidos como: oído, tacto y vista; es decir que reaccionan a cualquier efecto de manera muy similar a los estímulos de las personas y proporcionan información para activar o desactivar procesos. Existe una gran cantidad de sensores de los cuales se clasifican por su:

- ❖ Funcionamiento
- ❖ Señales
- ❖ Naturaleza de funcionamiento
- ❖ Elementos de fabricación

1.3.1. FUNCIONAMIENTO

- Activos.- requieren una Fuente de energía externa para su funcionamiento.
- Pasivos.- no requieren una Fuente de energía externa, más bien de las condiciones medioambientales para que se activen como se detalla en la Figura 1.4.

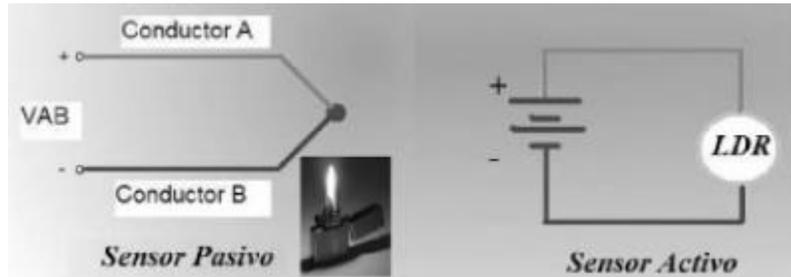


Figura. 1. 4 Clasificación según su funcionamiento
Fuente: (Ruiz, 2010)

1.3.2. SEÑALES

- Analógicos.- Pueden tomar información entre un sinnúmero de valores entre un mínimo y un máximo.
- Digitales.- Pueden tomar información explícitamente entre un 0 y 1 como muestra la Figura 1.5.

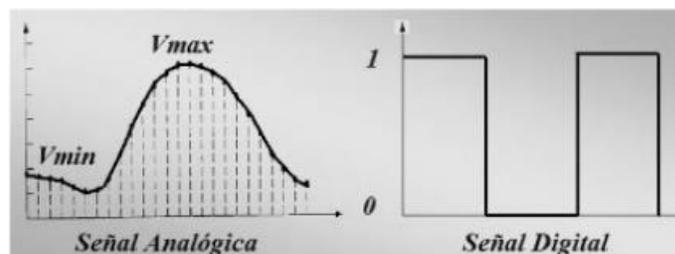


Figura. 1. 5 Clasificación según sus señales
Fuente: (Ruiz, 2010)

1.3.3. NATURALEZA DE FUNCIONAMIENTO

Los sensores se caracterizan por su funcionamiento que depende de varios factores como se puede observar en la Figura 1.6 y se detallan de la siguiente manera:

- Posición.- Experimentan variaciones según la posición de los elementos que lo componen.
- Fotoeléctricos.- Experimentan variaciones según la intensidad de luz que reciben.
- Magnéticos.- Experimentan variaciones según en campo magnético que los atraviesa.
- Temperatura.- Experimentan variaciones según la temperatura del lugar donde se encuentran.
- Humedad.- Experimentan variaciones según el nivel de humedad de donde se encuentren.
- Presión.- Experimentan variaciones según la presión a la que se someten.
- Movimiento.- Experimentan variaciones según al movimiento que se someten.
- Químicos.- Experimentan variaciones según los agentes químicos que inciden en ellos.



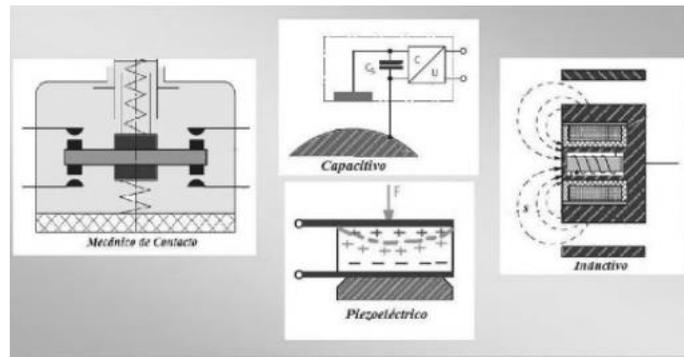
*Figura. 1. 6 Clasificación según su funcionamiento
Fuente: (Ruiz, 2010)*

1.3.4. ELEMENTOS DE FABRICACIÓN

En la elaboración de los sensores existen elementos que permiten cumplir con el correcto funcionamiento como se observa en la Figura 1.7 y se detallan a continuación:

- Mecánicos.- Utilizan contactos mecánicos que se accionan para abrir o cerrar.
- Resistivos.- Utilizan en su fabricación elementos resistivos.
- Capacitivos.- Utilizan en su fabricación condensadores.
- Inductivos.- Utilizan en su fabricación bobinas.
- Piezoeléctricos.- Utilizan en su fabricación cristales como el cuarzo.

- Semiconductores.- Utilizan en su fabricación semiconductores.



*Figura. 1. 7 Clasificación según su fabricación
Fuente: (Ruiz, 2010)*

1.4. SENSORES DE AGUA

Estos sensores cuentan con un alto poder de reconocimiento de nivel de agua gracias a sus pistas conductoras expuestas, cierran el circuito cuando detecten presencia de agua en ellas, se puede obtener datos que se utilizan para realizar actividades que dependen del valor que presente el sensor de agua, en la Figura 1.8 se observa las pistas conductoras.



*Figura. 1. 8 Sensor de Agua
Fuente: (Salazar & Fernandez, 2015)*

1.5. SENSORES DE GAS

Son dispositivos que detectan la presencia de gases combustibles y gases tóxicos, se pueden configurar para variar su precisión por medio de un calentador encargado de aumentar la temperatura al reaccionar a los gases y provoca un cambio en la resistencia interna.

El funcionamiento analógico facilita la implementación con cualquier microcontrolador para detectar GLP, propano, metano, alcohol, hidrógeno, humo. Se recomienda usar estos sensores en espacios confinados y pequeños debido a que su eficiencia es mayor. En la Figura 1.9 se detalla un sensor de alcohol.



*Figura. 1. 9 Sensor MQ3
Fuente: (Espinosa, 2010)*

1.6. SENSOR ACELERÓMETRO- GIROSCOPIO

Es un sensor que mide la aceleración producida por el movimiento. Su uso es muy común en la electrónica actual para detectar el desplazamiento de dispositivos tales como: teléfonos móviles, tabletas ipads o videoconsolas, ya que al conocer el sentido en el que se dirige, se puede saber la posición del sensor dentro de un espacio tridimensional.

Los acelerómetros internamente tienen un MEMS (Micro Electro Mechanical Systems) que de forma similar a un sistema masa-resorte permite medir la aceleración como se detalla en la Figura 1.10.

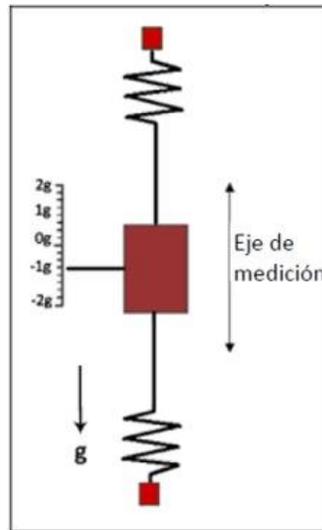


Figura. 1. 10 Funcionamiento Sensor Acelerómetro
Fuente: (Espinosa, 2010)

El giroscopio es un dispositivo que mide o mantiene el movimiento rotacional. Los giroscopios MEMS (sistema micro electromecánico) son sensores pequeños y baratos que miden la velocidad angular. Las unidades se miden en grados por segundo ($^{\circ}/s$) o revoluciones por segundo (RPS). La velocidad angular es la medición de la velocidad de rotación.

Además, un giroscopio MEMS de tres ejes, puede medir la rotación alrededor de tres ejes: x, y, z como se detalla en la Figura 1.11. Algunos giroscopios vienen en variedades de uno y de dos ejes, pero el giroscopio de tres ejes en un solo chip es el más útil.

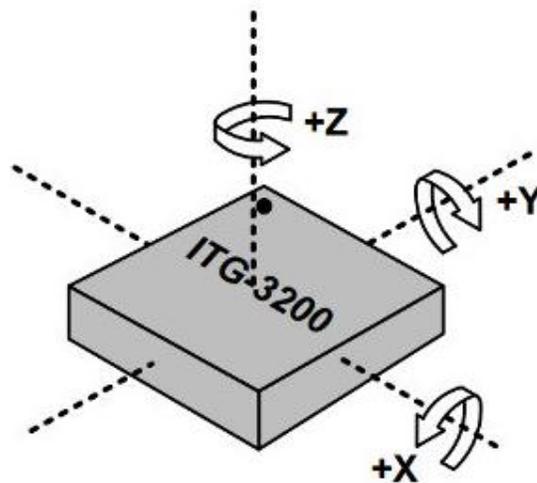


Figura. 1. 11 Giroscopio y sus ejes
Fuente: El Autor

1.7. SISTEMA GSM

Es un sistema celular digital muy utilizado, el sistema fue diseñado en un estándar europeo abierto lo que permitió resolver la interrupción del servicio cuando salen de su área de cobertura.

El elemento más importante en la red GSM (Global System for Mobile) es la identificación única de una tarjeta llamada SIM (Subscriber Identity Module). La tarjeta SIM no solo contiene datos básicos de identificación, sino también el número de identificación del suscriptor, claves de autenticación como también información de los servicios de telefonía prepago.

El principio de la conmutación empieza en el momento en que la llamada envía los datos de la señalización inicial donde el proceso más relevante es la selección de la estación móvil.

El terminal móvil envía el número de identificación IMSI (International Mobile Subscriber Identity) por medio de la estación base y del controlador de la estación base (BSC, Base Station Controller) que llega al centro de conmutación móvil MSC (Mobile Switching Centre).

El bloque AuC (Authentication Centre) envía a través del terminal móvil un número aleatorio que es transformado de acuerdo a los algoritmos y datos almacenados en la tarjeta SIM a otro número diferente que se envía como respuesta al número original para autenticar al usuario.

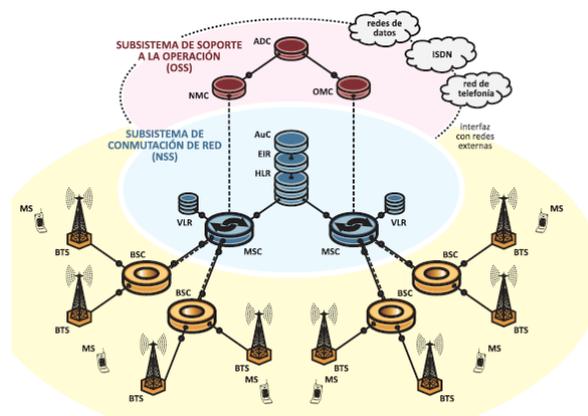
Posteriormente los datos individuales se comparan con los datos que hay en una base de datos ubicada en el bloque VLR. Si los datos son coherentes, se permite que el terminal móvil pueda acceder a la red móvil como muestra la Figura 1.12.

Para garantizar el anonimato, el terminal móvil se comunica con un número de identificación que se asigna provisionalmente, conocido como TMSI (Temporary Mobile Subscriber Identity). Gracias a ese número, el terminal móvil se identifica en el MSC.

Cuando la estación móvil cambia de estación base, también cambia de MSC, y se le reasigna otro TMSI. La transferencia de datos de usuario sólo puede comenzar una vez que estos procesos han terminado (Zdenek Becvar, 2016).

En el sistema GSM las funciones básicas trabajan en la banda de 900MHz por lo tanto el incremento de usuarios dio lugar a q se crearán otras versiones con diferentes frecuencias.

- GSM 900 – banda de frecuencias de 900 MHz, capacidad máxima de 2×124 canales, ancho de banda de 2×25 MHz
- GSM 1800 – banda de frecuencia de 1800 MHz, capacidad máxima de 2×374 canales, ancho de banda de 2×75 MHz
- GSM 1900 – banda de frecuencias de 1900 MHz, capacidad máxima de 2×298 canales, ancho de banda de 2×75 MHz



*Figura. 1. 12 Arquitectura del sistema GSM
Fuente: (Zdenek Becvar, 2016)*

1.8. SISTEMA GPS

El Sistema de Posicionamiento Global (GPS) es un sistema de radionavegación de los Estados Unidos de América, que se basa en el espacio y que proporciona servicios fiables de posicionamiento, navegación, y cronometría gratuita en todo el mundo.

Quien cuente con un receptor del GPS, el sistema le proporcionará su localización y la hora exacta en cualesquiera sean las condiciones atmosféricas, a cualquier hora y en cualquier lugar del mundo.

El GPS se compone de tres elementos: los satélites en órbita alrededor de la Tierra, las estaciones terrestres de seguimiento y control, y los receptores del GPS propiedad de los usuarios.

- Segmento espacial: El segmento espacial consta de al menos 24 satélites que vuelan alrededor de 20.200 kilómetros por encima de la Tierra con un período orbital de 11 horas y 58 minutos. Los satélites están organizados en seis órbitas con inclinación de 55°.
- Segmento de control: El segmento de control consta de instalaciones en tierra para el seguimiento, control y gestión de los satélites de segmento espacial. El núcleo del sistema de control es una estación principal de control, en Colorado en los Estados Unidos.
- Segmento de usuario: El segmento de usuario está representado por un receptor GPS, que procesa la información recibida de los satélites. El receptor GPS contiene un chip GPS con una componente radio, un procesador de señal digital, memoria, parte de control, y una interfaz a una unidad central. (GPS.GOV, 2012)

En base a la información que proporciona el GPS, es factible utilizar esa información en otros dispositivos electrónicos que brindan facilidades para la aplicación que se desee brindar, en la Figura 1.13 se detalla su estructura.



Figura. 1. 13 Sistema GPS
Fuente: (ANAINTE, 2014)

1.9. MINICOMPUTADORAS

Los Mini PCs son siempre una buena opción para disfrutar de toda la potencia de un computador, en un tamaño compacto. Se puede utilizar como servidor de contenidos, conectados al televisor, y por supuesto, como ordenador. Está formada por una placa que soporta varios componentes necesarios en un computador común, que es capaz de ser muy funcional.

En su funcionamiento, es necesario añadir una tarjeta de memoria micro SD para su almacenamiento y enchufarlo a la corriente gracias a cualquier cargador de tipo micro USB, además, en cuestión de su apariencia existen cajas predeterminadas que se adaptan al diseño de fabricación, como también se puede realizar la elaboración de una caja a gusto personal.

La fundación de Raspberry Pi pone a disposición desde su página web Raspbian, una distribución de Linux basada en Debian, pero también se puede recurrir a muchas de las distribuciones específicas que la comunidad de usuarios ha desarrollado para diversos fines.

En relación al modelo que escoja existe varias opciones de conexión, pero siempre tiene al menos un puerto de salida de video HDMI y otro de tipo RCA, mini Jack de audio y un puerto USB 2.0 al que se puede conectar un teclado o mouse, para ello es importante utilizar una Raspberry de dos puertos USB y sobre todo que exista un puerto RJ45 de conexión a internet como se detalla en la Figura 1.14.

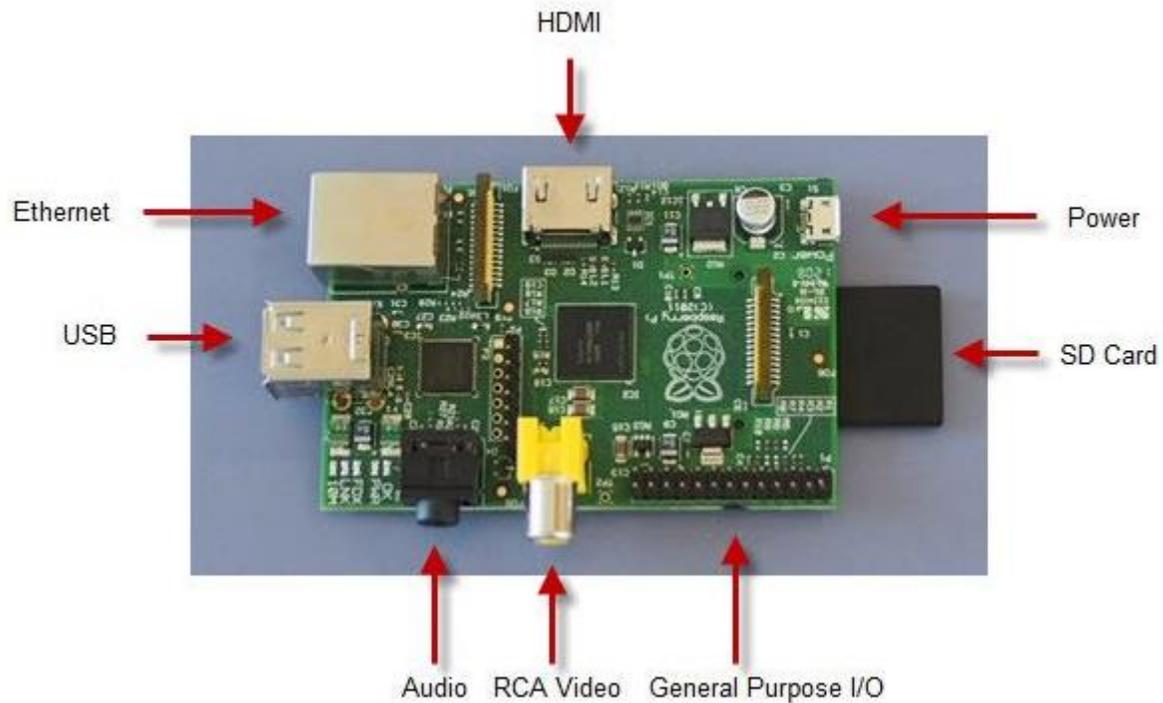


Figura. 1. 14 Minicomputador Raspberry
Fuente: (Gutierrez, 2017)

CAPÍTULO II

2. PROPUESTA

La caja negra es representada por un diagrama de bloques en la Figura 2.1, y su composición es de una etapa de control con la ayuda de un microcontrolador, que permite mantener el control de los sensores, actuadores y la etapa de comunicación del proyecto, que contiene un GPS NEO 6M y una SIM GSM800.

La etapa de control también permite que se active una cámara y un micrófono mediante una Raspberry Pi3 que se denomina como etapa de grabación, que al mismo tiempo facilita almacenar en una memoria micro SD la información que se genera en un evento y se identifica como etapa de almacenamiento.

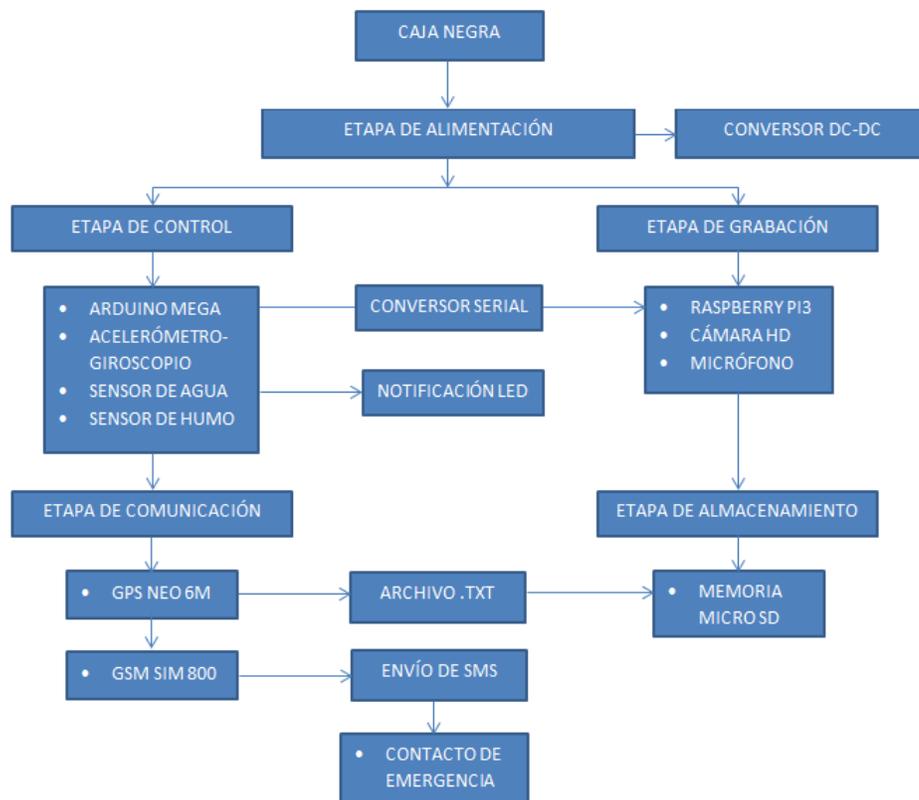


Figura. 2. 1 Diagrama Lógico de la Caja Negra
Fuente: El Autor

La caja negra se activa al detectar un cambio de velocidad repentino mediante un acelerómetro giroscopio o un conato de incendio detectado por un sensor de humo en el interior del vehículo, esta señal se envía inmediatamente a la Raspberry Pi3 para activar la cámara de video y un micrófono, que grabará por aproximadamente un minuto y almacena esta información en una memoria micro SD.

Al mismo tiempo la etapa de control envía una señal al módulo GSM para notificar por mensaje de texto a un contacto de emergencia que escoge el usuario.

En virtud de la señal emitida por la etapa de control se activa un GPS, que proporciona información de velocidad, posición en latitud y longitud, hora y fecha de cada vez que ocurre un evento; de esta información se genera en un archivo de formato .TXT y se almacena en la misma memoria micro SD. En la Figura 2.2 se presenta el diagrama de secuencia de los distintos elementos a ocupar.

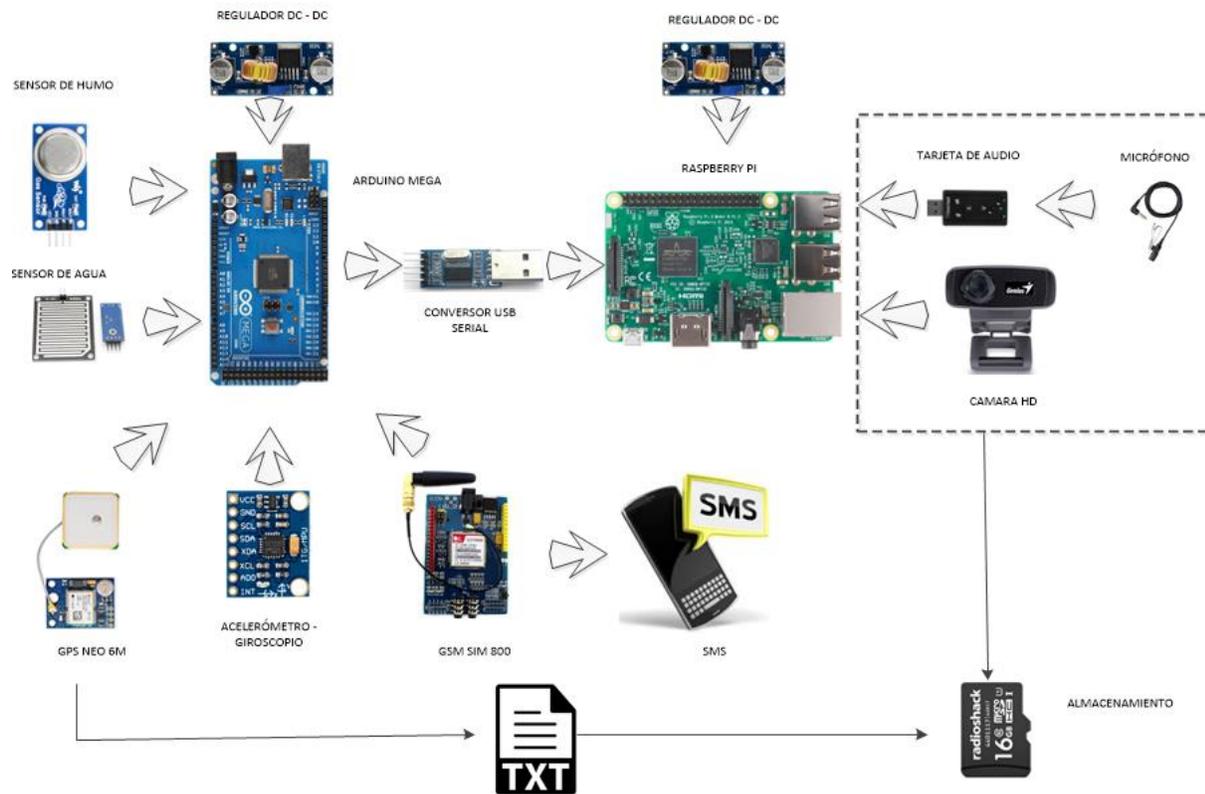
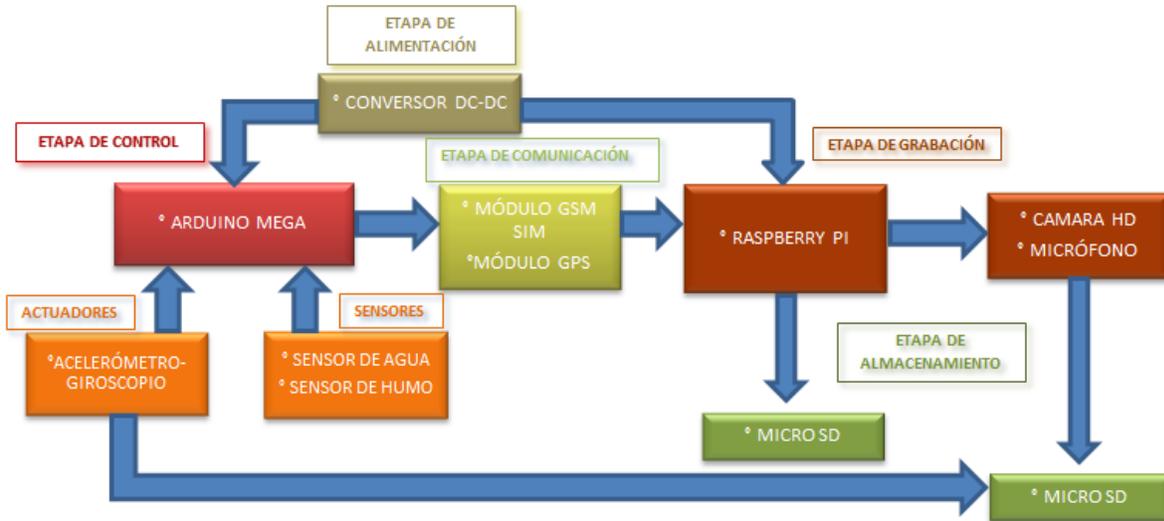


Figura. 2. 2 Secuencia de la Caja Negra
Fuente: El Autor

2.1. MÓDULOS DEL PROYECTO

En el interior del dispositivo se ha dividido por módulos de acuerdo al funcionamiento, los mismos que van a ejecutar tareas determinadas, en la Fig. 2.3 se detalla las etapas.



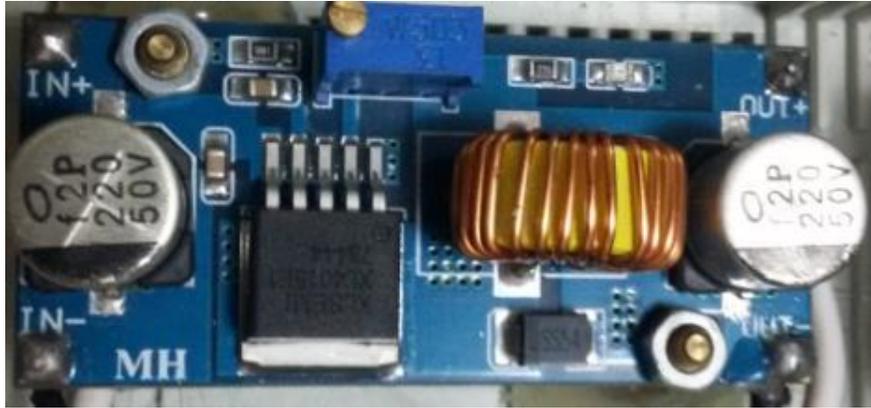
*Figura. 2. 3 Módulos de funcionamiento
Fuente: El Autor*

2.1.1. ETAPA DE ALIMENTACIÓN

El módulo de alimentación depende de una fuente de energía de 12 voltios que es suministrada de la batería del vehículo. Los voltajes que necesita la Caja Negra son de 3.3 V y 5 V, los cuales permitirán un adecuado funcionamiento de los elementos.

El convertor DC-DC XL4005 como el que se detalla en la Figura 2.4 es un regulador de tipo conmutador reductor con una alta eficiencia de conversión y excelente regulación de línea. Permite obtener un voltaje regulado a partir de una fuente máxima de 32V DC y mínima de 1.5V DC.

La alimentación con el convertor DC-DC se usa de manera independiente para que permita distribuir las cargas de la caja negra que superan los 5 A, así el regulador del Arduino y de la Raspberry Pi manejaran cargas inferiores a los 4 A independientemente.



*Figura. 2. 4 Regulador convertidor DC-DC
Fuente: (ROBOSHOP, 2015)*

2.1.2. ETAPA DE CONTROL

El módulo de control del proyecto se encarga de integrar los elementos que lo componen, en función de recibir la información de los actuadores y sensores, los datos obtenidos se enviarán al Arduino Mega.

Este módulo funciona con un Arduino Mega que contiene un microcontrolador Atmega 2560 como se detalla en la Figura 2.5, donde se procesa las diferentes señales de los elementos que lo componen, así también se verifica las conexiones de las entradas y salidas de cada uno de los pines que ejecutan las acciones configuradas y así permitir la comunicación con la Raspberry Pi.



*Figura. 2. 5 Arduino Mega
Fuente: (ARDUINO, 2016)*

En la selección del módulo Arduino se realiza una comparación de las principales características de los elementos más comerciales y disponibles en el mercado electrónico, como se detalla en la Tabla 2.1.

Tabla. 2. 1 Tipos de Arduino

Característica de Arduino	UNO	Mega 2560	Leonardo	DUE
Tipo de microcontrolador	Atmega 328	Atmega 2560	Atmega 32U4	AT91SAM3X8E
Velocidad de reloj	16 MHz	16 MHz	16 MHz	84 MHz
Pines digitales de E/S	14	54	20	54
Entradas analógicas	6	16	12	12
Salidas analógicas	0	0	0	2 (DAC)
Memoria de programa (Flash)	32 Kb	256 Kb	32 Kb	512 Kb
Memoria de datos (SRAM)	2 Kb	8 Kb	2.5 Kb	96 Kb
Memoria auxiliar (EEPROM)	1 Kb	4 Kb	1 Kb	0 Kb

Fuente: (ARDUINO, 2016)

De los elementos comparados anteriormente, se utiliza un Módulo Arduino Mega por su capacidad de salidas de comunicación, ya que la caja negra se dimensiona con 3 pines de comunicación.

2.1.3. ETAPA DE ACTUADORES

Los actuadores en la caja negra se encargan de recibir señales, estas deben configurarse para cumplir las necesidades del proyecto, que por medio de éstas activarán los procesos de grabación y comunicación. En este módulo se encuentra como actuador al ACELERÓMETRO-GIROSCOPIO.

2.1.3.1. ACELERÓMETRO/ GIROSCÓPIO

El giroscopio es un dispositivo que funciona para medir velocidades angulares, basado en el mantenimiento del impulso de rotación. Este dispositivo muestra el cambio de rango de rotación en sus ejes X, Y y Z como en la Tabla 2.2

Tabla. 2. 2 Rangos de operación

Rango de Escala Completa Giroscopio	Sensibilidad del Giroscopio	Rango de Escala Completa Acelerómetro	Sensibilidad del Acelerómetro
(+-250)	131	(+-2)	16384
(+-500)	65,5	(+-4)	8192
(+-1000)	32,8	(+-8)	4096
(+-2000)	16,4	(+-16)	2048

Fuente: (Lara Eduardo, 2015)

El Acelerómetro mide la aceleración, inclinación o vibración y transforma la magnitud física de aceleración en una magnitud eléctrica que será la que se emplea en los equipos de adquisición estándar como detalla la Figura 2.6. Éstos se aplican para cada eje X, Y y Z al igual que en la velocidad angular que mediante las señales obtenidas por el mismo, envía una señal al compilador arduino el cual se comunica con el módulo de Raspberry Pi3, que se encarga de procesar estos datos y enviar un SMS de alerta al número de contacto móvil configurado en el módulo GSM SIM cada vez que ocurra un cambio brusco de velocidad o posición.



Figura. 2. 6 Acelerómetro- Giroscopio

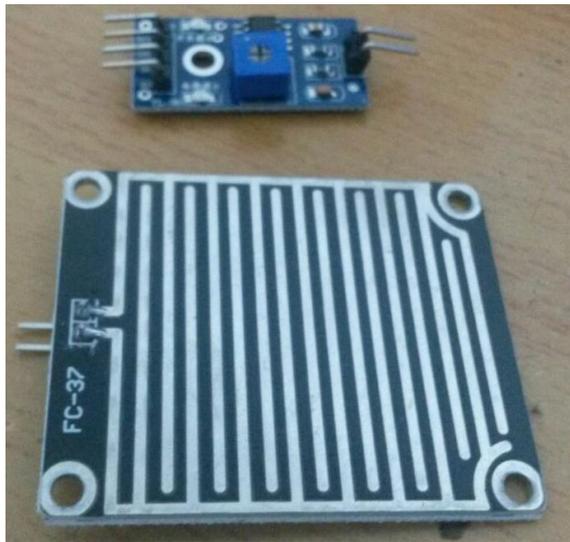
Fuente: (Ingeniería MCI Ltda., 2004)

2.1.4. ETAPA DE SENSORES

Los sensores son receptores muy similares a los reflejos humanos, estos se encargan de recibir una señal que permite activarlos para emitir datos que ejecutan otra función en un mismo proceso. Los sensores que se ocupan en el desarrollo del proyecto son: sensor de agua y sensor de humo.

2.1.4.1. SENSOR DE AGUA

El sensor de agua de la Figura 2.7 genera señales de acuerdo a la cantidad de agua que se ubique en el dispositivo y se acciona al juntar las pistas conductoras por medio del agua que producen ciclos cerrados, que envía una señal al compilador arduino e indica por medio de un LED que la calzada esta mojada y que el conductor debe tomar precaución al conducir sobre superficies mojadas.



*Figura. 2. 7 Sensor de Agua
Fuente: (García, 2105)*

2.1.4.2. SENSOR DE HUMO

El dispositivo de la Figura 2.8 es regulado en su sensibilidad, que se ajusta con una carga resistiva entre los pines de salida y tierra con la finalidad de obtener un dato más preciso a su primer evento. Se conecta 5V a cualquiera de los pines (A o B) y el sensor empieza a emitir tensión. Este a su vez envía información del dato obtenido al Arduino Mega, para emitir una señal de alerta sonora y al envío de un SMS a un número de contacto móvil que pueda socorrer en la emergencia del vehículo en presencia de humo.



*Figura. 2. 8 Sensor de Humo
Fuente: (STEREN, 2015)*

2.1.5. ETAPA DE COMUNICACIÓN

Este módulo depende de las etapas de control y grabación, los mismos que se comunican mediante un conversor USB serial como muestra la Figura 2.9. Este conversor se encarga de recibir la información que proporciona la etapa de control y a su vez permite activar la cámara y el micrófono por medio de la Raspberry Pi3.



*Figura. 2. 9 Conversor USB Serial
Fuente: (ROBOSHOP, 2015)*

2.1.5.1. GSM SIM 800

El módulo GSM SIM 800 permite al Arduino Mega utilizar la red celular para el envío de SMS cuando este tenga algún tipo de notificación con respecto a los eventos como: volcamiento, choque o conato de incendio. En la Figura 2.10 se detalla un módulo GSM de la etapa de comunicación, que se alimenta de la fuente externa del convertor DC-DC del módulo de control.



Figura. 2. 10 Módulo GSM SIM 800
Fuente: (Salazar & Fernandez, 2015)

2.1.5.2. GPS NEO 6M

Este equipo viene equipado en el PCB con una EEPROM de configuración de fábrica, una pila de botón que mantiene los datos de configuración en la memoria EEPROM, un indicador LED y una antena cerámica como la Figura 2.11. También posee los pines o conectores Vcc y Gnd que son de alimentación y los pines Rx y Tx que sirven de comunicación, medio por el que está conectado un microcontrolador mediante una interfaz serial. Así permite que el GPS pueda funcionar en sus parámetros iniciales con el fin de comunicarse hacia el satélite y entregar su posición con un margen de error de 4 a 8 metros.



Figura. 2. 11 GPS NEO 6M
Fuente: (GPS.GOV, 2012)

2.1.6. ETAPA DE GRABACIÓN

Permite almacenar datos de audio y video de un sistema administrado por la Raspberry Pi3 como detalla la Figura 2.12. Este módulo se activa al recibir una señal del Arduino Mega por cada evento sucedido.



*Figura. 2. 12 Conexión de una Raspberry Pi3
Fuente: El Autor*

2.1.6.1. RASPBERRY PI3

Al encender la Raspberry Pi se pone en marcha automáticamente y comienza a arrancar el sistema operativo, mostrando en pantalla toda una serie de líneas de código. En pocos segundos después se puede acceder a la interfaz gráfica de Raspbian y a su modo de configuración correspondiente. Aquí se carga la configuración deseada para que pueda grabar tanto audio como video a partir de los eventos generados en el Arduino Mega.

La mini PC de 4 puertos USB, un puerto HDMI y un puerto Ethernet, se alimenta de 5V DC, en los puertos USB se conecta un convertor USB serial ara la comunicación del Arduino Mega y un HUB USB que permite conectar la cámara HD, una tarjeta de audio USB y una tarjeta de memoria micro SD, que sirve de almacenamiento para los eventos como se puede identificar en la Figura 2.13



*Figura. 2. 13 Raspberry y Conversor USB Serial
Fuente: El Autor*

2.1.6.2. CÁMARA HD

En la validación de la cámara es muy importante verificar la calidad de la misma, y que permita visualizar en una resolución de vídeo CIF / VGA hasta 30 fps (fotogramas por segundo), 720P HD hasta 30 fps, Interfaz USB 2.0 con una imagen fija de resolución 1MP, 1280 x 720, 640 x 480 píxeles con un archivo en formato MJPEG / WMV UVC (Plug & Play) como la Figura 2.14. Únicamente se activa cuando detecte un frenazo, volcamiento o que el sensor detecte presencia de humo, es decir un evento.



*Figura. 2. 14 Cámara HD Genius
Fuente: (Genius, 2015)*

2.1.6.3. MICRÓFONO

La finalidad de grabar audio en un percance automovilístico, lo determina la posibilidad de obtener una información audible, de todas maneras, el micrófono de la Figura 2.15, se activa con las mismas características que la cámara y se accionan al mismo tiempo, esto permite generar archivos de grabación que se puedan reproducir con facilidad.



*Figura. 2. 15 Micrófono
Fuente: (Genius, 2015)*

La principal característica de utilizar un micrófono multimedia, es el modo de conexión de la Raspberry Pi3 mediante una tarjeta de audio que facilita una mayor precisión en su operación.

2.1.7. ETAPA DE ALMACENAMIENTO

La unidad de almacenamiento se verá utilizada en cada instante donde se produzca un evento, para ello se ha determinado que se guarden los eventos de manera independiente como en la Tabla 2.3

Tabla. 2. 3 Almacenamiento de Archivos

CARPETA VIDEO	CARPETA AUDIO	CARPETA DE DATOS
Grabación de Video	Grabación de Audio	Ubicación
		Hora
		Fecha
		Velocidad

Fuente: El Autor

2.1.7.1. MICRO SD

En la unidad de almacenamiento se coloca dos memorias externas Micro SD, con capacidad de almacenamiento de 16GB, en una memoria se almacena la configuración. Mientras que la otra unidad de almacenamiento se utiliza para archivar los eventos en una carpeta con un archivo de formato .TXT; la información de video es compatible con el reproductor VLC, las unidades de almacenamiento son extraíbles como se observa el adaptador de lectura Micro SD en la Figura 2.16.



Figura. 2. 16 Adaptador de Lectura Micro SD

Fuente: El Autor

2.2. ASPECTOS TÉCNICOS DEL PRODUCTO EN SU TOTALIDAD

El dispositivo cuenta con un convertor DC-DC de una alimentación de 12V, que también permite trabajar con voltajes menores de los 32V, la corriente que entrega el convertor es de 5A.

El Arduino Mega tiene varias salidas de transmisión y recepción, que permite conectar los actuadores y sensores. También es importante recalcar que la Raspberry Pi3 debe tener una tarjeta de almacenamiento de 16 GB ya que los programas cargados a su funcionamiento cubren los 8.1 GB, por ende, se necesita una memoria de mayor capacidad.

El GSM SIM 800 permite trabajar en GPRS (General Packet Radio Service) que es una extensión mejorada de la red GSM a una velocidad de transferencia de 85,6 Kbps y la tensión de alimentación es de 3,4 V.

El GPS NEO 6M tiene un reloj incorporado y trabaja de manera simultánea con los datos que proporciona al momento de generar un evento, en donde sus datos serán reflejados en latitud y longitud con un rango de referencia a la ubicación de +/- 4 a 8 metros.

La etapa de almacenamiento del dispositivo dispone de 55 Horas y 46 Minutos, que aproximadamente genera un archivo de 10 MB por cada evento generado, lo que determina que puede existir una grabación constante por 2 días y 7 horas aproximadamente.

El dispositivo cuenta con un sensor de agua FC-37 permite recibir valores entre 0 y 1023 unidades, estos dependen de la cantidad de agua que cierren las pistas conductoras del sensor y calibrar su nivel de sensibilidad.

El sensor de humo MQ2 permite obtener un valor de funcionamiento que depende de la concentración de humo que recepte, este valor se puede configurar al controlar la sensibilidad de operación para activar otras actividades.

El dispositivo cuenta con una cámara HD de 720p para una mejor percepción del video y se almacena en una tarjeta de memoria micro SD.

2.3. ANÁLISIS DE COSTOS

Previo a realizar el análisis de costos, se establece un presupuesto, como límite de implementación se ha fijado 600 USD (seiscientos dólares estadounidenses).

En la tabla 2.4 se detallan los materiales que se necesitan en la implementación del proyecto, al ser estos los más necesarios, se detalla un presupuesto tentativo mediante la adquisición de elementos y dispositivos por separado.

Tabla. 2. 4 Lista de materiales para construcción del proyecto

DESCRIPCION	VALOR UNITARIO	CANTIDAD	VALOR TOTAL
Raspberry PI 3	\$58,00	1	\$58,00
Micro SD 16 Gigas	\$16,50	2	\$33,00
Adaptador USB a micro SD	\$5,00	1	\$5,00
Adaptador de Audio a USB	\$12,00	1	\$12,00
Convertidor USB a serial	\$4,01	1	\$4,01
Cámara Webcam USB HD	\$35,00	1	\$35,00
Micrófono	\$10,00	1	\$10,00
Cable adaptador de USB a USB	\$12,00	1	\$12,00
Arduino MEGA	\$20,50	1	\$20,50
Módulo GPS NEO 6M	\$26,31	1	\$26,31
Acelerómetro - Giroscopio	\$8,77	1	\$8,77
Sensor de Agua	\$6,47	1	\$6,47
Sensor de Humo MQ2	\$16,00	1	\$16,00
Regulador DC-DC	\$9,46	2	\$18,92
Módulo SIM GSM 800	\$27,00	1	\$27,00
Hub USB	\$15,00	1	\$15,00
Protoboard 2 regletas	\$15,00	1	\$15,00
Pulsador	\$0,08	3	\$0,24
Baquilita 20x30	\$10,00	1	\$10,00
Caja	\$14,37	1	\$14,37
Materiales Extras	\$70,00	1	\$70,00
		TOTAL	\$417,59

Fuente: El Autor

Una vez determinado los materiales necesarios y estimado un presupuesto, se realiza el análisis de 3 opciones, dos opciones mediante las proformas (Anexos) y la tercera opción mediante la adquisición de los elementos como detalla en el Tabla 2.5.

Tabla. 2. 5 Descripción de cotizaciones

PRESUPUESTO. EVALUACION DE OPCIONES			
OPCIONES	Opciones para la implementación	Costo	Observaciones
Opción 1	Compra de la mayoría de elementos	\$ 417,59	Opción más costosa con la mayoría de implementos en un solo lugar.
Opción 2	Proforma 2. Compra de elementos sin Raspberry, Módulo GPS, Módulo GSM SIM, Acelerómetro-Giroscopio y Reguladores DC-DC	\$ 112,65	Opción más económica sin la mayoría de implementos importantes.
Opción 3	Proforma 2. Compra de elementos sin Raspberry, Módulo GPS, Módulo GSM SIM Y Acelerómetro- Giroscopio	\$ 113,05	Opción adecuada, aunque costaría más comprar los implementos individualmente

Fuente: El Autor

Realizado el análisis, se concluye que la Opción 1 es la más adecuada ya que se encuentran la mayoría de implementos necesarios e indispensables, por ende, la Opción 2 y la Opción 3 no serán tomadas en cuenta por la escases de elementos indispensables en la construcción del proyecto, de ser tomadas en cuenta es necesario dedicar más tiempo para obtener el resto de implementos.

Con los antecedentes mencionados se discriminaron las proformas por el aspecto de factor de tiempo, la variación de los precios en las proformas no es alta, pero sí muy importante la falta de implementos necesario en el proyecto.

En relación al tiempo establecido, se realiza un breve estudio para determinar el costo horas-hombre, se considera un sueldo base de 1 salario mínimo por tratarse de un estudiante que aún no ha obtenido el título de tercer nivel.

Salario mínimo unificado 2018 = USD 386 x 1 = USD 386 correspondientes a 160 horas laborales.

USD 386/160 horas = USD 2,41 por cada hora.

Este es el valor con el cual se realiza el costo hora-hombre, se muestra en la tabla 2.6 de la siguiente manera.

Tabla. 2. 6 Costo horas Hombre

HORAS/DIA	DIAS/SEMANA	SEMANAS	COSTO/HORA	PERSONA	TOTAL
3	4	11	2,41	1	318,12

Fuente: El Autor

Para el proyecto también se considera el tema de imprevistos que es un valor que no se puede excluir, en este caso se considera aplicar un 10% del presupuesto antes determinado, debido a los gastos que no se ha considerado por temas de movilización, alimentación durante la ejecución del proyecto, posibles daños en los elementos que mayormente se manipulan, consumo de energía eléctrica, etc. Por lo tanto, queda un valor de USD 60. Por ende, se determina un costo total de \$795,71 los mismos que se detalla en la Tabla 2.7.

Tabla. 2. 7 Monto de inversión

COSTO TOTAL INVERTIDO		
ELEMENTO	INVERSION	COSTO
Opción 3	Directa	\$ 417,59
Horas-Hombre	Directa	\$ 318,12
Imprevistos	Indirecta	\$ 60,00
TOTAL		\$ 795,71

Fuente: El Autor

2.4. TIEMPO REQUERIDO

En primer lugar se fija los elementos a utilizar en el diseño y construcción de la Caja Negra para vehículos, razón por la cual es muy importante investigar qué tipos de módulos y tarjetas permiten realizar la una comunicación entre los elementos, la búsqueda se la realiza vía internet con documentos y tesis que hayan realizado un trabajo similar al planteado en el tema de tesis, la búsqueda ayuda a determinar qué tipo de elementos utilizará en el proyecto, y si es compatible con lo planteado inicialmente.

Se determina utilizar dos dispositivos importantes en la elaboración del proyecto, que son el módulo Arduino Mega y la Raspberry Pi3, que permite una comunicación estable gracias a la librería de Arduino.

En la siguiente etapa se analiza el sistema de comunicación entre los sensores y actuadores con el Módulo Arduino Mega, en donde se necesita verificar los estados de trabajo de cada elemento con los resultados, esto permite definir que pines de conexión se utiliza en la comunicación de los elementos, esta etapa incorpora al microcontrolador Atmega2560.

En la comunicación del Acelerómetro-Giroscopio, se necesita verificar los parámetros que permiten identificar la posición del dispositivo según el plano tridimensional, para esto se comprueba los valores que se obtienen cuando el dispositivo se encuentra en posición horizontal, vertical e inclinado; con la finalidad de configurar el actuador con los datos que se mencionan. Esta información facilita la notificación del acelerómetro-giroscopio hacia al Arduino Mega y del arduino al módulo GSM SIM, para que éste envíe un SMS del evento generado.

También, se realizan pruebas de otra etapa en el proceso de funcionamiento; como es el envío y recepción del SMS por medio de un módulo GSM SIM 800 que se adapta al Arduino Mega, este SMS refleja la información si el dispositivo detecto un volcamiento del vehículo, choque o empezó a detectar humo en la cabina. Cabe recalcar que el dispositivo también envía un SMS para notificar que la caja negra se inicia en su funcionamiento, con la finalidad de probar la conexión del módulo SIM al encender el dispositivo.

Con la Raspberry Pi3 se necesitan los puertos USB para conectar los elementos que permiten grabar los eventos generados, también se realizan pruebas de verificación por medio de una memoria micro SD de los eventos grabados como audio y video. Cada uno de los puertos USB utilizados en la Raspberry Pi deben ser etiquetados por la asignación y reconocimiento de comunicación del periférico de salida.

La etapa de armado del circuito en un protoboard permite comprobar el funcionamiento del dispositivo electrónico, para así verificar las fallas existentes y realizar las correcciones respectivas.

En la adaptación del circuito y de sus elementos en una caja plástica, se debe organizar la distribución interna de cada uno de ellos, así se facilita la adaptación de los elementos que tienen periféricos de salida, se considera que existen sensores y actuadores que se adaptan a la parte exterior de vehículo; para lo cual se realizan sus conexiones con terminales de mayor sujeción. Los orificios en el acrílico se crearon con una máquina de corte laser que permite mejorar la precisión en adaptación de los elementos.

Para todo este proceso se considera el siguiente cronograma, con las tareas asignadas de acuerdo a los avances que se tiene programado como detalla el diagrama de GANTT en la Figura 2.17.

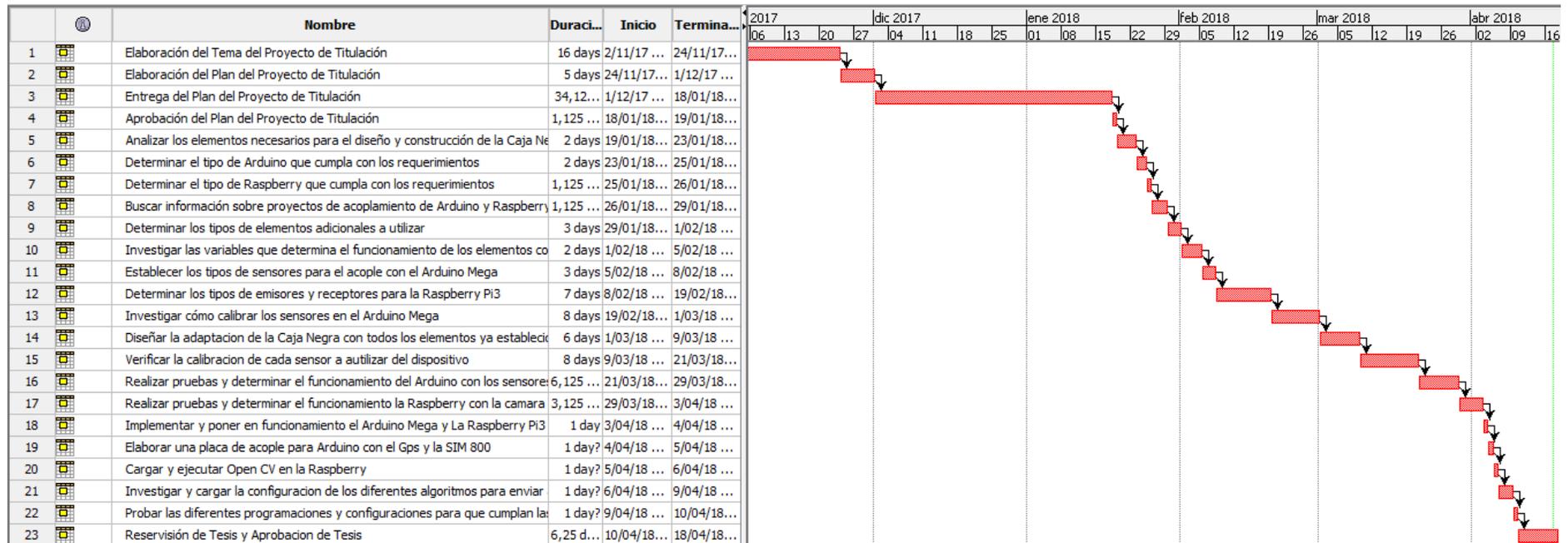


Figura. 2. 17 Diagrama de GANTT
Fuente: El Autor

CAPÍTULO III

3. IMPLEMENTACIÓN

La implementación del proyecto, se encarga de plasmar las ideas planteadas con la justificación técnica de cada uno de los elementos, a su vez, la construcción del dispositivo se adapta a cada uno de los módulos, como son: la etapa de alimentación, etapa de control, etapa de comunicación, etapa de grabación y la etapa de almacenamiento.

3.1. DESARROLLO

Para elaborar la caja negra se ha considerado trabajar en base al siguiente diagrama, el cual explica los procesos de funcionamiento.

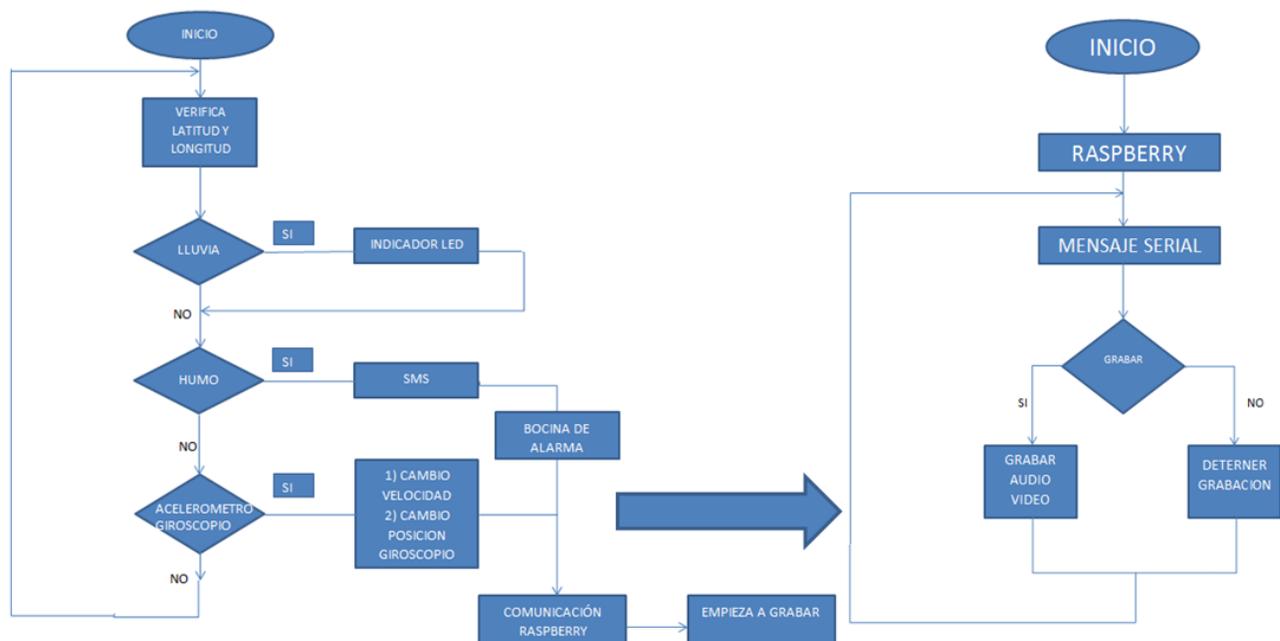


Figura. 3. 1 Diagrama de Bloques de la Caja Negra
Fuente: El Autor

La figura 3.1 representa de forma general la Caja Negra mediante un diagrama en bloques, dónde se establece las partes principales; el funcionamiento se basa en el movimiento del vehículo que contiene su interior al dispositivo, el cual verifica el estado del sensor de agua y se activa un indicador LED cuando los contactos se cierran, mediante esta información se deduce que la calzada en la que se desplaza el vehículo está mojada. A su vez el sensor de humo realiza una validación, si existe humo en la cabina se activará para notificar mediante una bocina y un mensaje de texto que hay presencia de un conato de incendio en el interior del vehículo.

Una vez analizado los estados de los sensores cuando el dispositivo se encuentra en movimiento y se produce un cambio de velocidad de manera brusca, notificará mediante un SMS al número de contacto móvil configurado en la tarjeta SIM, que incorpora la etapa de comunicación.

Cuando ocurra un evento también se activará la etapa de grabación de audio y video, conjuntamente con la posición en latitud y longitud del GPS, además almacena la velocidad, hora y fecha en un mismo archivo de la memoria micro SD.

3.2. DISEÑO

Para el desarrollo de la Caja Negra se ha implementado el diseño con los elementos más importantes que permiten el funcionamiento del producto.

En el diseño del dispositivo se coloca un indicador LED para notificar que el sensor de agua se activó, en la adaptación es necesario colocar una resistencia que protege el funcionamiento del indicador, a continuación, se presenta los siguientes cálculos.

$$V = 5 V$$

$$I = 0,015 A$$

$$R = \left(\frac{V}{I} \right)$$

$$R = \left(\frac{5}{0,015} \right)$$

$$R = 333 \Omega$$

En la Caja Negra, como se muestra en la Figura 3.2; se consideran 5 etapas de funcionamiento: etapa de alimentación, etapa de control, etapa de comunicación, etapa de grabación y etapa de almacenamiento.

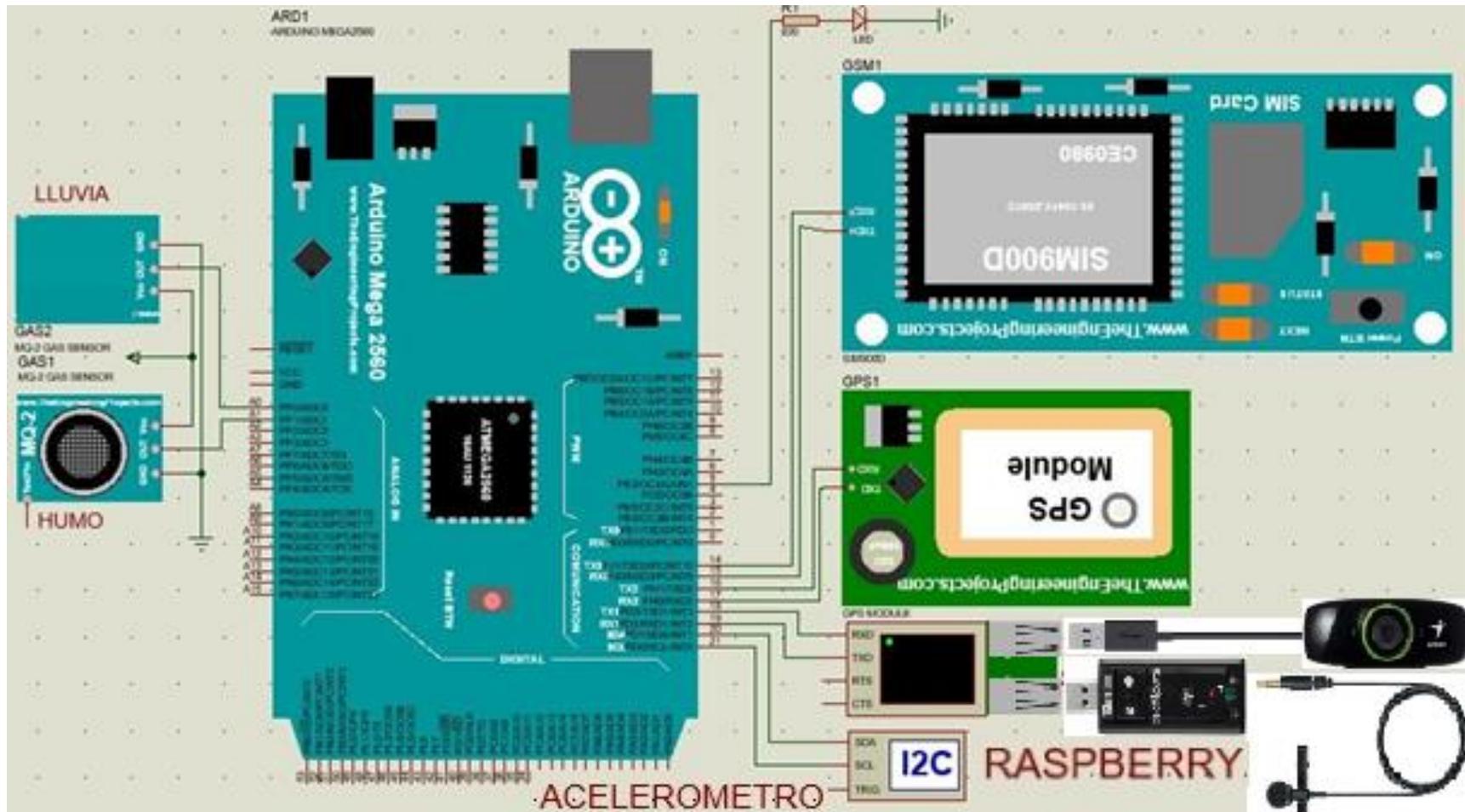


Figura. 3. 2 Diseño del dispositivo electrónico para almacenamiento de información caja negra
Fuente: El Autor

3.2.1. ALIMENTACIÓN DE LOS MÓDULOS ARDUINO Y RASPBERRY

La distribución de las cargas que se utiliza para justificar la alimentación con dos convertidores DC-DC, se determina mediante un balance de cargas en los módulos principales, como se verifica en la Tabla 3.1 del Módulo Arduino Mega.

Tabla 3. 1 Distribución de cargas Módulo Arduino Mega

CARGA CONVERTOR 1	
ARDUINO MEGA	0,5 A
MÓDULO GSM SIM	1,2 A
MÓDULO GPS NEO	0,4 A
ACELERÓMETRO GIROSCÓPIO	0,5 A
SENSOR DE HUMO	0,1 A
SENSOR DE AGUA	0,02 A
TOTAL	2,72 A

Fuente: El Autor

Para las cargas dimensionadas del Arduino Mega se puede instalar un regulador de 3 A, donde lo más recomendable es tener un valor de tolerancia del 20% de las cargas, que comprende a un valor de 0,54 A, por ende, exige un regulador DC-DC superior equivalente a 5 A.

En la distribución de cargas del Módulo RaspberryPi3, se considera la Tabla 3.2 que indica las cargas de funcionamiento.

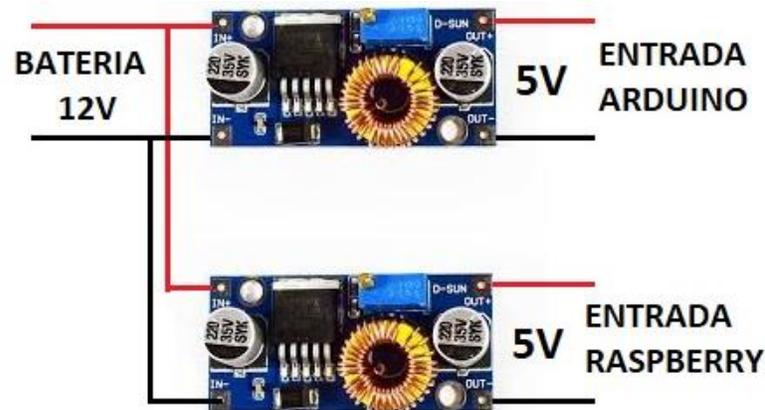
Tabla 3. 2 Distribución de cargas Módulo Raspberry

CARGA CONVERTOR 2	
RASPBERRY	2,4 A
CAMARA	0,4 A
MICROFONO	0,05 A
CONVERTOR USB SERIAL	0,07 A
TOTAL	2,92 A

Fuente: El Autor

Con la distribución de cargas del Módulo Raspberry lo más recomendable es colocar un regulador de 5 A, para cubrir un valor de tolerancia del 20% que representa a 0,58 A de las cargas del módulo.

Esta etapa se relaciona con el voltaje que se adapta tanto al módulo Arduino como a la Raspberry Pi, el voltaje de entrada es de 12V, para lo cual se colocó dos reguladores de voltaje DC-DC que permitan obtener un voltaje de salida de 5V como detalla en la Figura 3.3.



*Figura. 3. 3Diagrama de conexión de convertor DC-DC a 5V
Fuente: El Autor*

3.2.2. DISEÑO DE LA PLACA PCB PARA LA ADAPTACIÓN DE LOS ELEMENTOS

En la adaptación de los implementos que comprenden la etapa de control, es necesario diseñar una placa PCB que permita interconectar cada uno de los elementos, como son los sensores, actuadores, módulos y alimentación de los dispositivos como se detalla en la Figura 3.4.

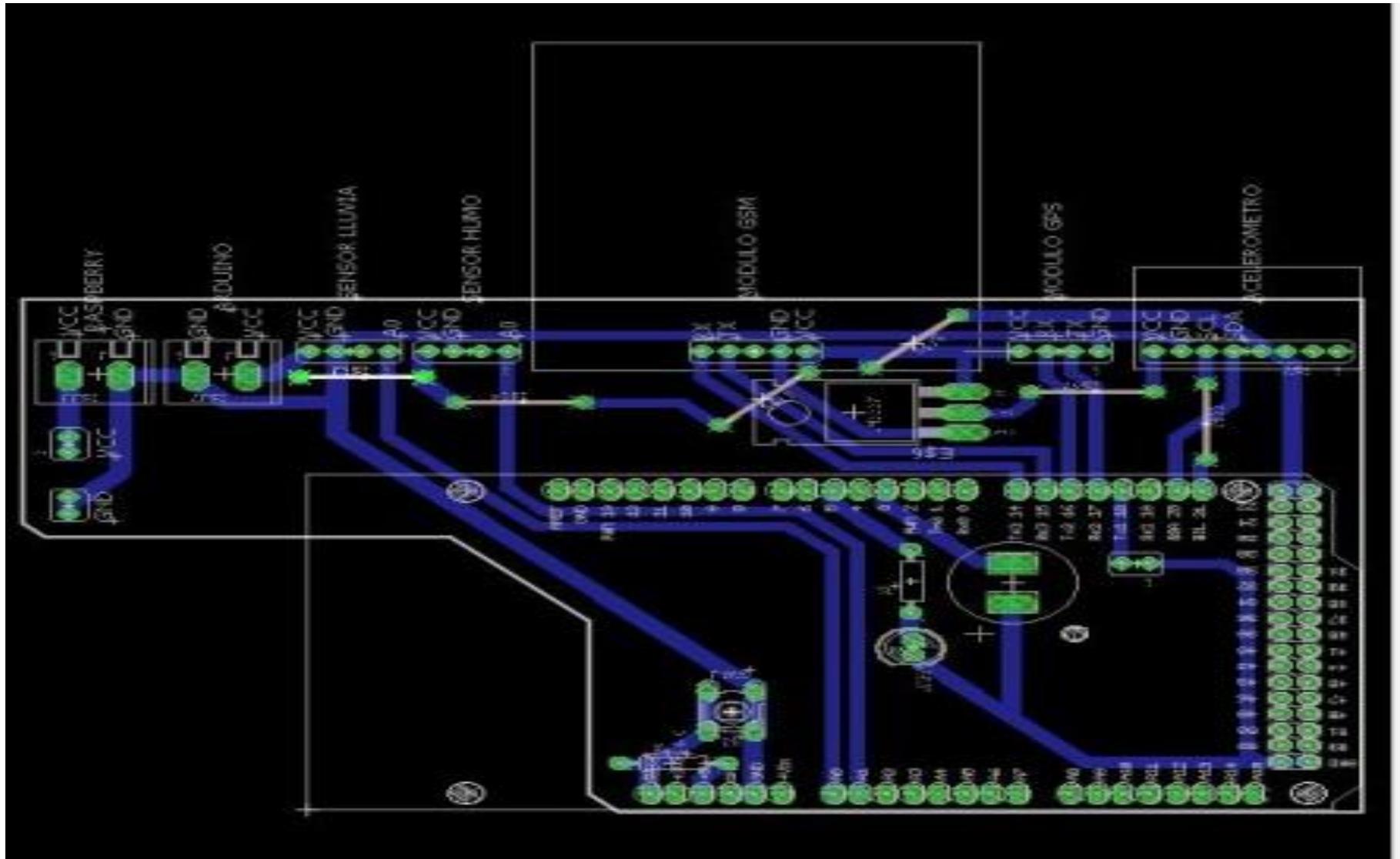


Figura. 3. 4 Diseño de placa PCB
Fuente: El Autor

En la distribución de la placa PCB se considera colocar una parte para alimentar a la Raspberry como al Arduino Mega.

3.2.3. CONEXIÓN DE LOS MÓDULOS DE LA ETAPA DE CONTROL

En la etapa de control se debe obtener un Arduino Mega que es la parte primordial del funcionamiento, donde permite procesar las señales de los sensores, actuadores y elementos de comunicación.

El sensor de agua dispone de 3 pines, donde 2 corresponden a su alimentación y un tercer pin que facilitará la comunicación entre el Arduino y el sensor como muestra la Figura 3.5.

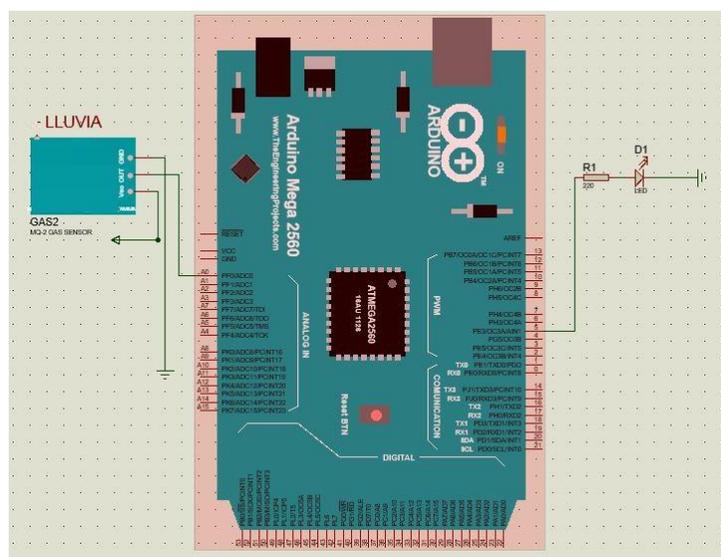


Figura. 3. 5 Diagrama de conexión del Sensor de Agua
Fuente: El Autor

Para obtener una medida precisa y oportuna de la notificación del sensor, es importante calibrarlo, las unidades en las que oscilan los parámetros de medición son las que se detallan en la Tabla 3.3; estos valores se reflejan en la configuración del software.

Tabla 3. 3 Rango de configuración de Sensor de Agua

SENSOR DE AGUA	RANGO DE OSCILACIÓN	VALOR CONFIGURADO
	0 - 1023 unidades	< 350 unidades

Fuente: El Autor

En la conexión del sensor de humo MQ2 se identifica los pines que realizan la comunicación con el Módulo Arduino, en este punto se conectan las entradas analógicas que dispone el módulo, como también los pines de alimentación del sensor que se verifica en la Figura 3.6.

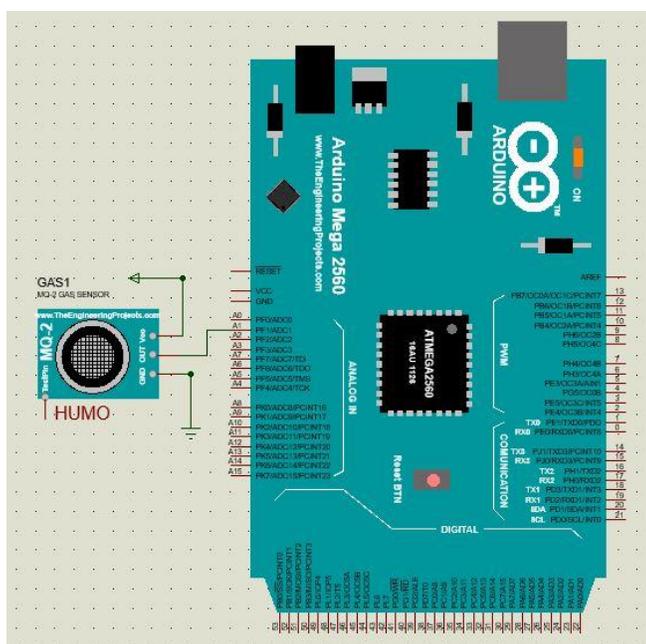


Figura. 3. 6 Diagrama de conexión del Sensor de Agua

Fuente: El Autor

En la calibración del sensor de Humo se verifican los parámetros de medición que se detallan en la Tabla 3.4; estos valores van a estar configurados en el software del dispositivo.

Tabla 3. 4 Rango de configuración de Sensor de Humo

SENSOR DE HUMO	RANGO DE OSCILACIÓN	VALOR CONFIGURADO
	300 – 10000 ppm	> 650 ppm

Fuente: El Autor

La conexión del Módulo GSM SIM 800 depende de dos pines que permite realizar la comunicación con el Módulo Arduino, se debe identificar los pines de transmisión y recepción para conectarlos de manera inversa. Así permitirá una correcta comunicación como detalla la Figura 3.7.

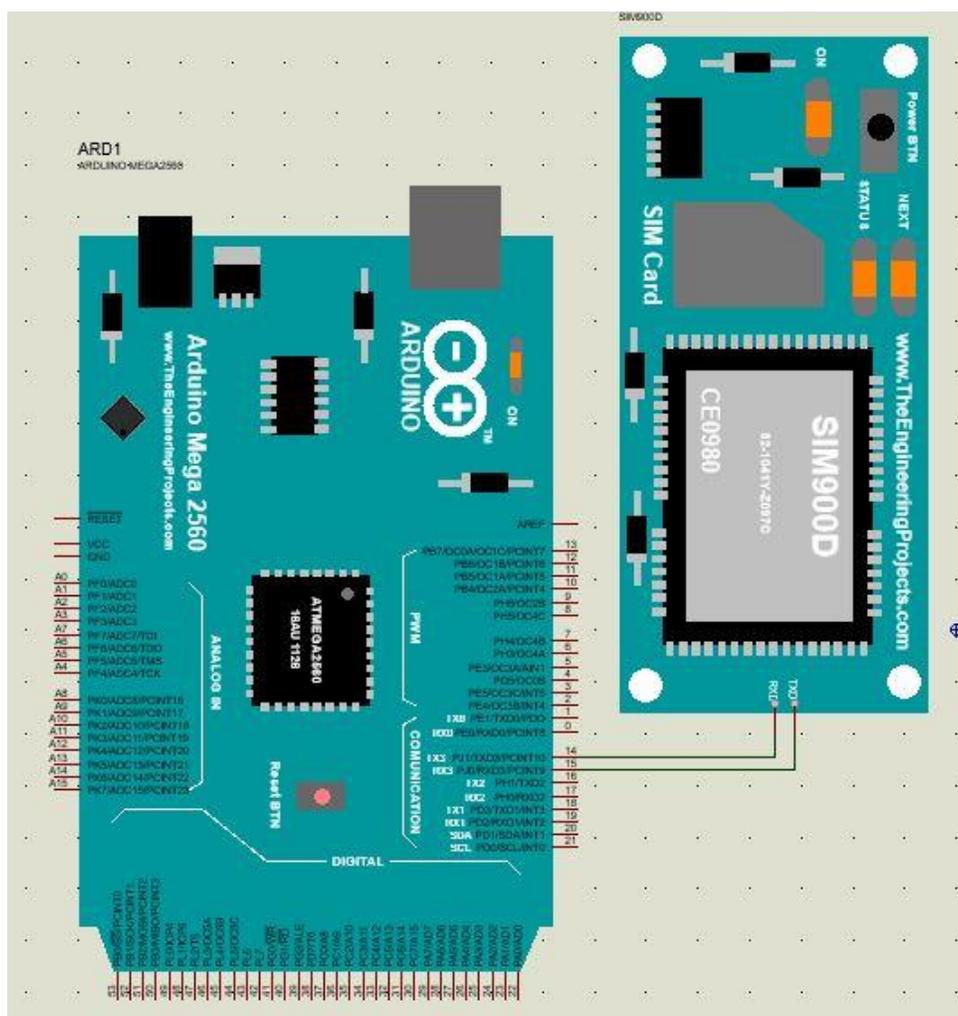


Figura. 3. 7 Diagrama de conexión del Módulo GSM SIM 800
Fuente: El Autor

En la comunicación entre el Módulo GPS NEO 6M y el Módulo Arduino Mega, se verifica la conexión en los terminales de trasmisión y recepción, donde se toma en cuenta que el Módulo Arduino Mega dispone de varios pines para este tipo de conexiones como detalla en la Figura 3.8.

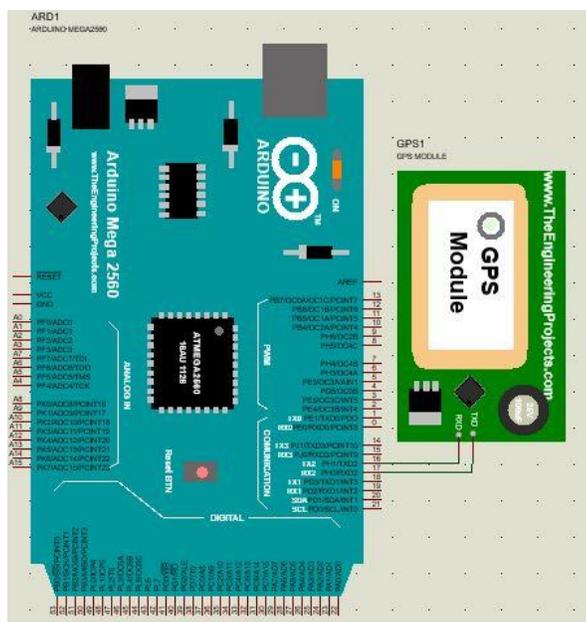


Figura. 3. 8 Diagrama de conexión del Módulo GPS NEO 6M
Fuente: El Autor

Para la verificación de la posición del vehículo, se necesita el Módulo Acelerómetro – Giroscopio. Este módulo dispone de pines SDA (Serial Data) y SCL (Serial Clock) que se debe conectar de manera paralela con el Módulo Arduino Mega, que simplifica una comunicación segura y digitalizada como muestra en la Figura 3.9.

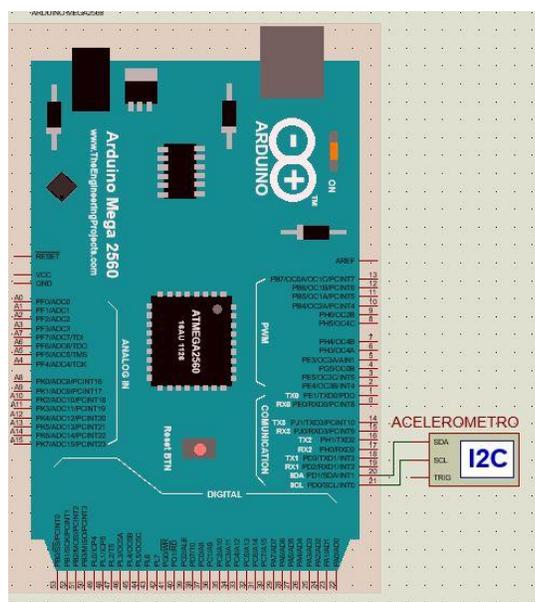
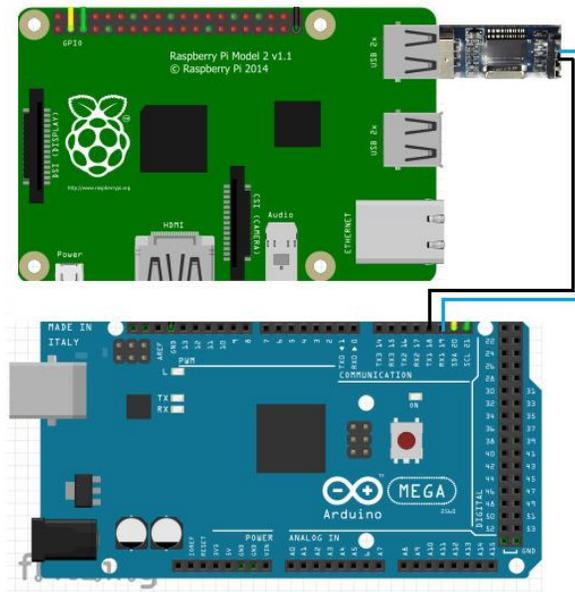


Figura. 3. 9 Diagrama de conexión del Módulo Acelerómetro-Giroscopio
Fuente: El Autor

3.2.4. CONEXIÓN DE LA ETAPA DE COMUNICACIÓN

Es necesario utilizar un acople entre el Módulo Arduino Mega y la Raspberry Pi, que permita intercomunicarse para procesar la información obtenida de los sensores que se adaptan al Arduino, esta comunicación genera un envío de información mediante un convertor USB serial como detalla en la Figura 3.10.



*Figura. 3. 10 Diagrama de conexión del Módulo Arduino Mega y Raspberry Pi
Fuente: El Autor*

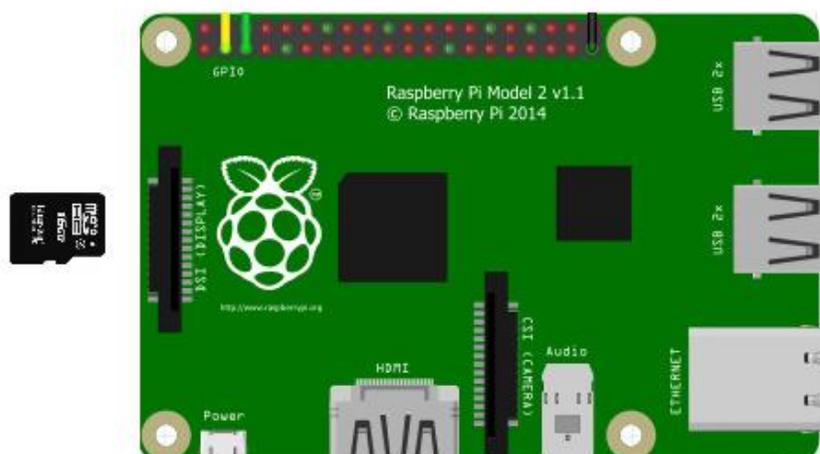
Una parte muy importante en la etapa de grabación son los medios que se utilizan para grabar audio y video, en los cuales se realiza una conexión de la Raspberry Pi a un Hub USB, que permite obtener los periféricos de salida disponibles con mejor facilidad en conexión para la cámara y el micrófono, como detalla en la Figura 3.11.



*Figura. 3. 11 Diagrama de conexión de la Raspberry Pi con la Cámara y Micrófono
Fuente: El Autor*

3.2.5. ELABORACIÓN DE LA ETAPA DE ALMACENAMIENTO

En la etapa de almacenamiento se guarda la información del programa que se ejecuta en la Raspberry Pi, este medio de almacenamiento es por una tarjeta de memoria Micro SD de 16GB, que se introduce en la parte posterior de la Raspberry Pi como detalla en la Figura 3.12.



*Figura. 3. 12 Diagrama de conexión de la tarjeta de almacenamiento del programa de la Raspberry Pi
Fuente: El Autor*

Para la capacidad de memoria que se necesita en el almacenamiento de los programas de la Raspberry Pi, se realiza una tabla con las especificaciones necesarias de los programas como se observa en la Tabla 3.5.

Tabla 3. 5 Dimensión de Memoria para Raspberry

DESCRIPCIÓN	PESO
Sistema Operativo cargado	4 GB
Open Cv	3 GB
Otros Programas	0,1 GB
Juego de Instrucciones de Actualización	1 GB
TOTAL	8,1 GB

Fuente: El Autor

Por tal motivo se necesita instalar una tarjeta de memoria Micro SD de 16 GB para cubrir la demanda de memoria del Módulo Raspberry Pi.

En el almacenamiento de información del GPS NEO 6M mediante el Módulo Arduino Mega, se necesita una tarjeta de memoria Micro SD de 16 GB, adaptada en un lector de memoria Micro SD, que comparte el espacio de la tarjeta con los archivos que se generan en la etapa de grabación, como representa en la Figura 3.13.

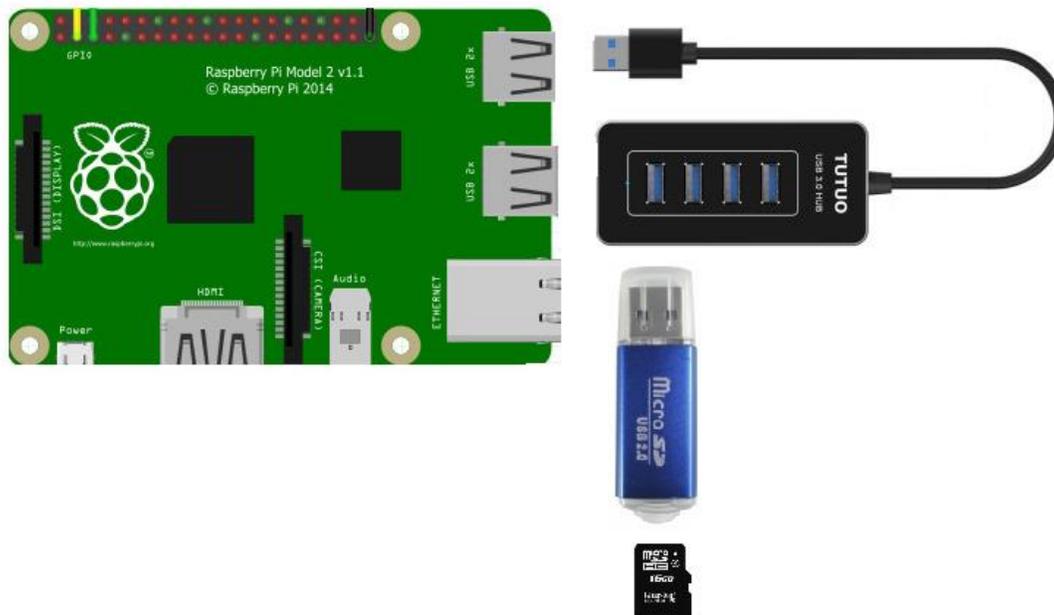


Figura. 3. 13 Diagrama de conexión de la tarjeta de almacenamiento de información

Fuente: El Autor

La capacidad de almacenamiento de la tarjeta de memoria para guardar la información se determina de la siguiente manera ver Tabla 3.6:

Tabla 3. 6 Tamaño de archivos en almacenamiento

ITEM	TIEMPO	AUDIO	VIDEO	ARCHIVO .TXT
1	0:01:06	5,6 MB	4,2 MB	0.00033 MB
2	0:01:05	5,4 MB	3,7 MB	0.00033 MB
3	0:01:03	5,3 MB	3,5 MB	0.00033 MB
TOTAL	0:03:14	16,3 MB	11,4 MB	0,001 MB
PROMEDIO	0:01:05	5,43 MB	3,8 MB	0.00033 MB
PROMEDIO GENERAL	9,23 MB			

Fuente: El Autor

La identificación en relación al tamaño que dispone cada archivo se la realizó en muestra de 3 archivos almacenados, en donde se puede identificar que para un evento ocurrido se ocupa 10MB de su almacenamiento.

Los 10 MB hacen relación a que se puede almacenar hasta 1600 eventos en una tarjeta de almacenamiento Micro SD.

La tarjeta de memoria del dispositivo de 16GB proporciona un almacenamiento de 55 horas y 46 minutos, se debe tomar en cuenta que cada archivo que se genera será aproximadamente de 10 MB.

3.2.6. DISEÑO DE LA CAJA DEL PROYECTO

Para el diseño y dimensión de la caja, se necesita distribuir internamente cada uno de los implementos en una caja plástica con las dimensiones que se representan en la Figura 3.14.

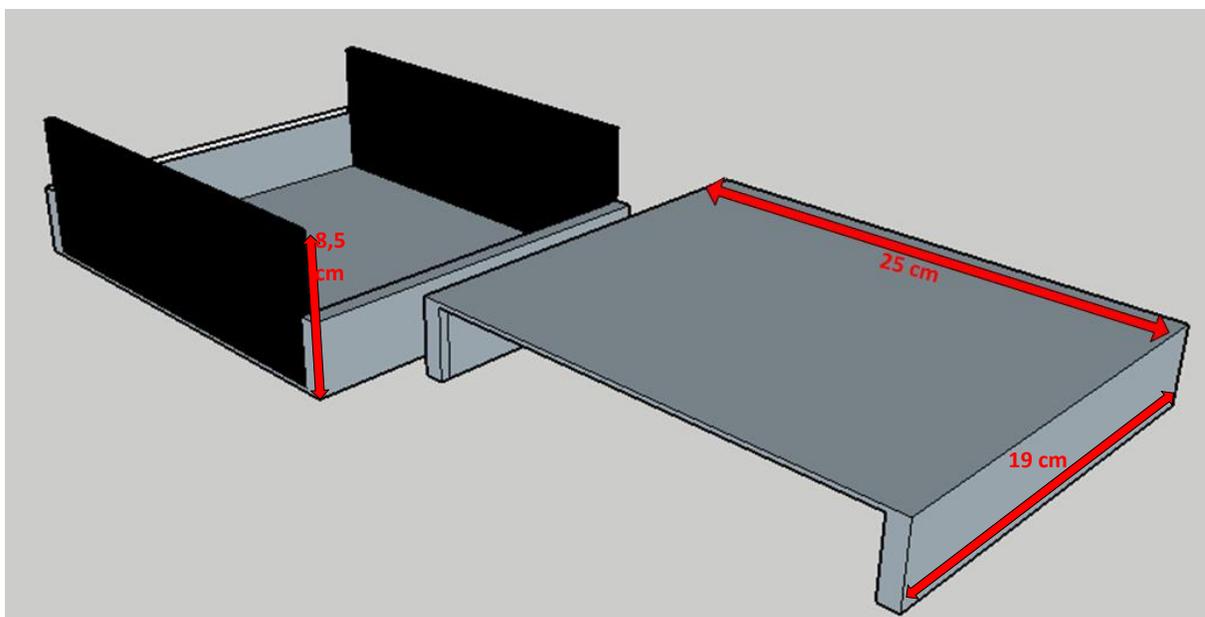


Figura. 3. 14 Diseño Mecánico de Caja Plástica
Fuente: El Autor

Las dimensiones de cada uno de los elementos permiten ordenar y distribuir de manera adecuada cada uno de ellos, así también facilita las conexiones para la implementación del proyecto como se detalla en la figura 3.15

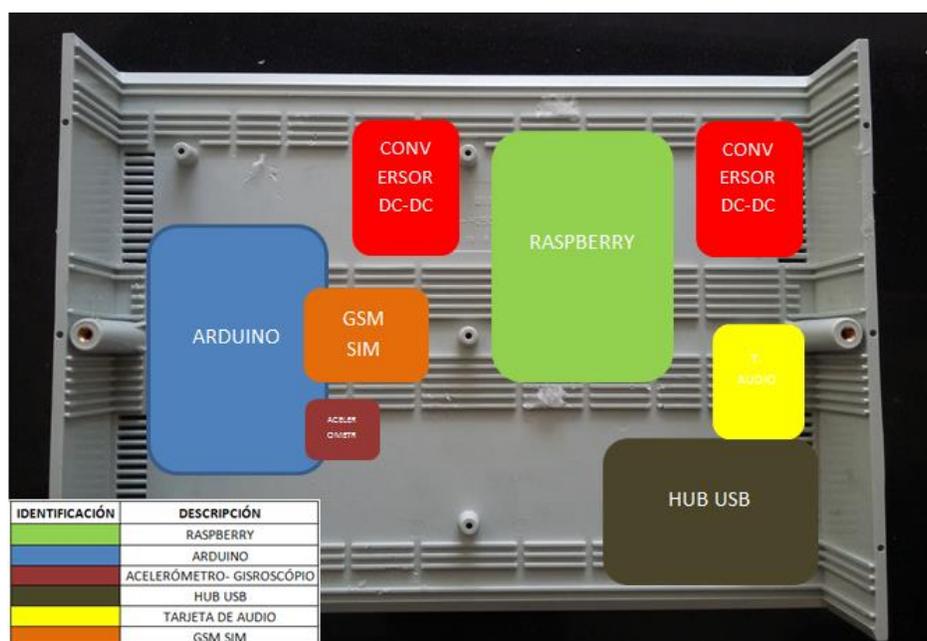
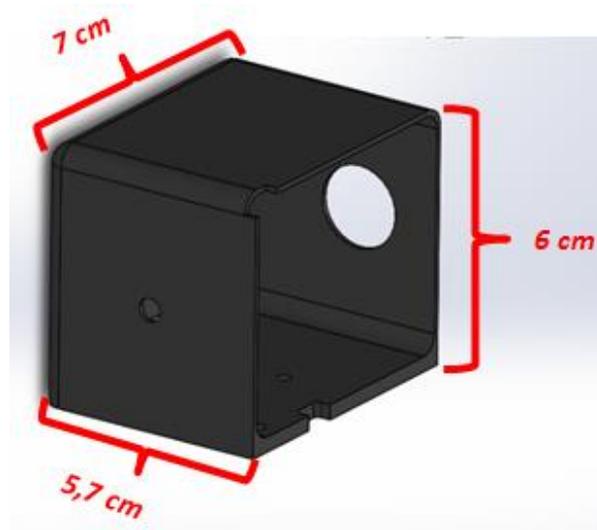


Figura. 3. 15 Distribución de elementos
Fuente: El Autor

En el diseño y adaptación de un elemento que sostenga al Módulo GPS NEO 6M y la Cámara HD es necesario realizar una caja a medida, con las especificaciones necesarias que permitan adaptar los elementos con facilidad como detalla en la Figura 3.16.



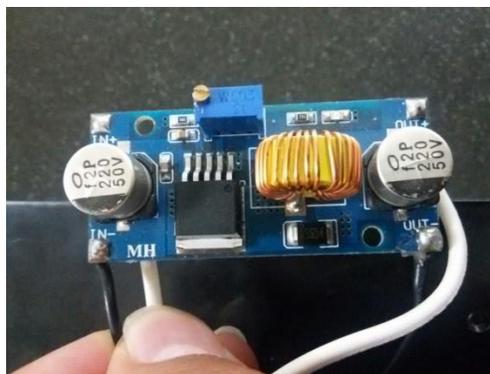
*Figura. 3. 16 Diseño Mecánico para la Cámara y GPS
Fuente: El Autor*

3.3. IMPLEMENTACIÓN DEL PROYECTO

Para la construcción del proyecto se realiza la conexión y montaje de todos los elementos que lo componen en su implementación.

3.3.1. ETAPA DE ALIMENTACIÓN

En la conexión e implementación de la etapa de alimentación se verifica las entradas y salidas de voltaje del módulo, como en la Figura 3. 17, estos dos módulos están conectados en paralelo con una fuente de entrada de 12V DC que es la alimentación a todo el proyecto.



*Figura. 3. 17 Conversor de voltaje DC-DC
Fuente: El Autor*

Estos convertidores se encargan de regular el voltaje a 5V DC para alimentar al Módulo Arduino Mega y el otro convertor alimentará a la Raspberry Pi como se verifica en la Figura 3.18.



*Figura. 3. 18 Colocación del conversor de voltaje DC-DC
Fuente: El Autor*

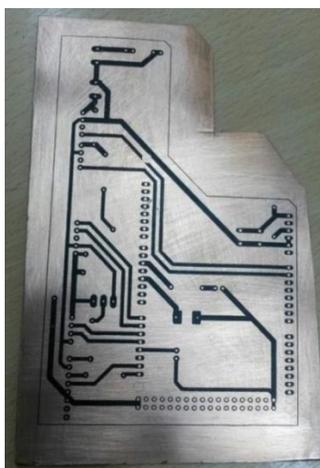
3.3.2. ETAPA DE CONTROL

Para la etapa de control se instala el Módulo Arduino Mega como indica la Figura 3.19, en donde la organización que se necesita en el interior de la caja es muy importante, se debe considerar que los implementos que van adaptados en el Arduino Mega deben quedar distribuidos sobre los pines que alimentan su funcionamiento.



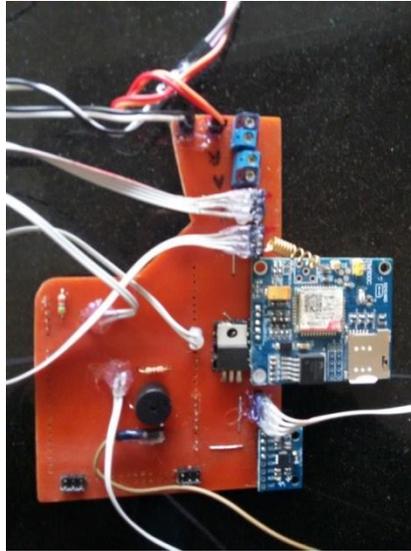
*Figura. 3. 19 Colocación del Módulo Arduino Mega
Fuente: El Autor*

Al realizar la conexión de los implementos, se verifica el diseño de donde quedan distribuidos cada uno de los elementos, que a su vez tienen que estar organizados en la placa PCB para una mejor distribución, y sobre todo una buena línea de conexión que simplifique las pistas como se detalla la Figura 3.20.



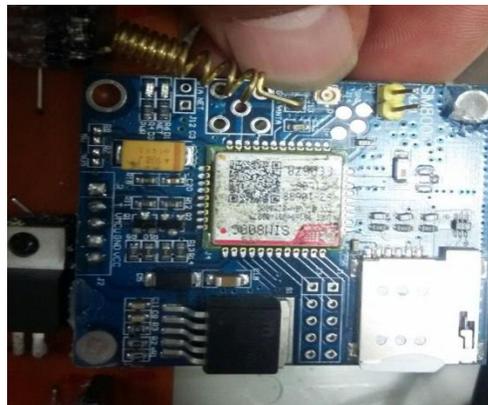
*Figura. 3. 20 Impresión de placa PCB
Fuente: El Autor*

La distribución de los elementos facilita conectar los pines del Módulo Arduino Mega con la placa PCB, esto permite que los pines de conexión de cada uno de los implementos queden expuestos para que se acople al Arduino Mega, como se detalla en la Figura 3.21



*Figura. 3. 21 Adaptación de los elementos a la placa PCB
Fuente: El Autor*

La conexión de los módulos GSM y la placa PCB permiten la conexión con el Módulo Arduino Mega, como la Figura 3.22. Donde se identifican los pines de Tx (Transmisión) y Rx (Recepción) que tiene en ambos módulos, a su vez estos se conectan de manera inversa para generar una conexión de recepción y transmisión.



*Figura. 3. 22 Adaptación de Modulo GSM SIM 800 a la placa PCB
Fuente: El Autor*

En la conexión del Módulo GPS Neo 6M se considera utilizar un bus de datos de 4 vías como muestra en la Figura 3.23.



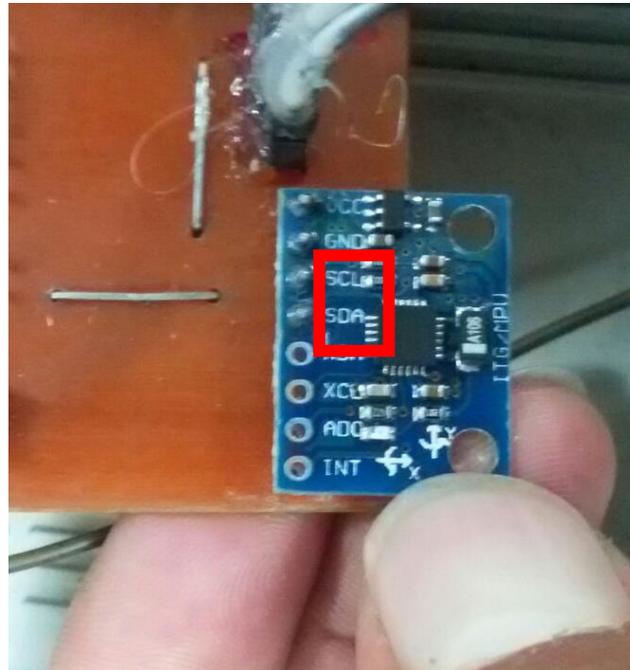
*Figura. 3. 23 Conexión de Bus de datos de 4 Vías
Fuente: El Autor*

De los cuales 2 son considerados para alimentar el Módulo y las otras dos vías se utilizan en la transmisión y recepción del GPS como detalla la Figura 3.24.



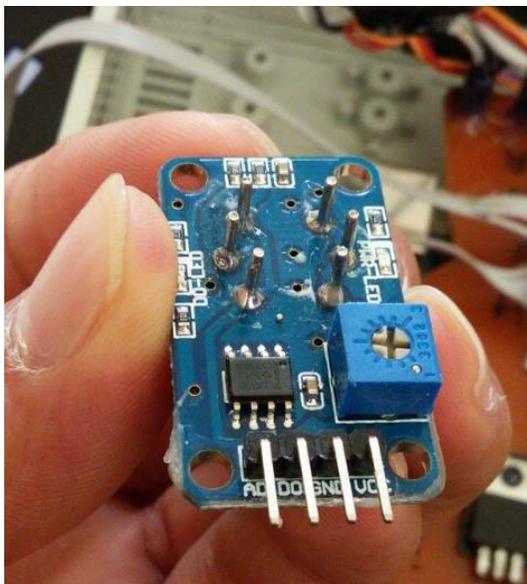
*Figura. 3. 24 Conexión de Bus de datos para Tx y Rx
Fuente: El Autor*

En la instalación del Acelerómetro Giroscopio, se considera utilizar 4 pines del dispositivo como muestra en la Figura 3.25, estos 2 pines corresponden a la alimentación del dispositivo como GND y un VCC. También se optimiza la comunicación entre el Módulo arduino Mega y el acelerómetro mediante los pines SDA (Serial Data) y SCL (Serial Clock) que también dispone el Arduino Mega.



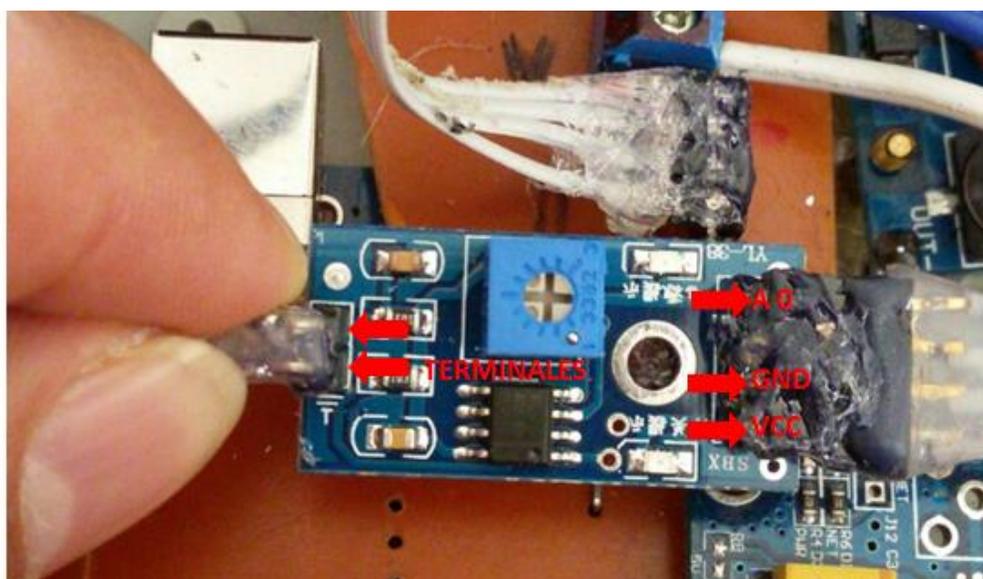
*Figura. 3. 25 Conexión de Acelerómetro-Giroscopio en placa PCB
Fuente: El Autor*

En esta adaptación de un Sensor de Humo MQ2 con el Arduino Mega, se consideran 3 pines, de los cuales dos pertenecen a la alimentación del dispositivo que son GND y VCC; el otro pin A0, hace realización a la señal que se genera al momento de receptor humo en el dispositivo. El cual envía la señal por ese pin, que indica al Arduino Mega que debe accionar la bocina y a la vez que envíe una señal al GSM SIM 800 para notificar mediante mensaje de texto que existe humo donde se encuentra el dispositivo.



*Figura. 3. 26 Identificación de pines del Sensor de Humo
Fuente: El Autor*

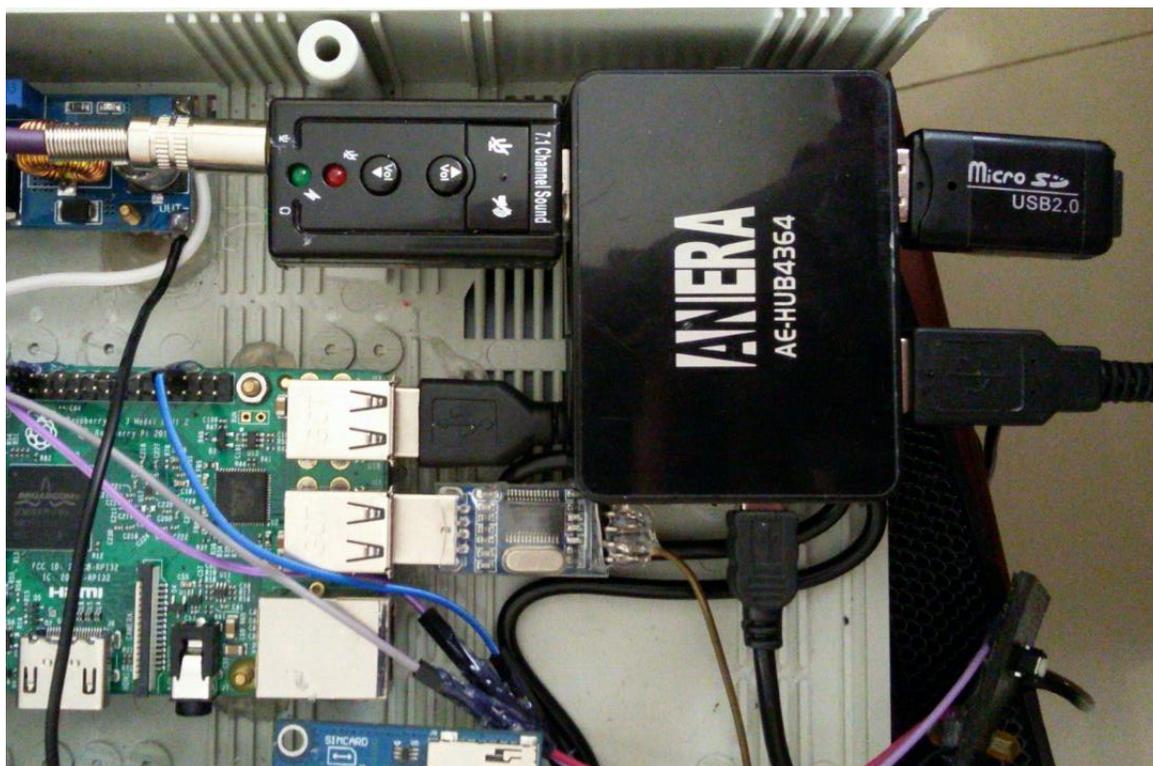
En la conexión del sensor de agua mediante la PCB, dispone de las entradas de GND y VCC para su alimentación, como detalla la Figura 3.27, también dispone de un pin A0 que envía una señal cuando los terminales del sensor se unen, es decir que realiza un corto circuito.



*Figura. 3. 27 Conexión terminales del Sensor de Agua
Fuente: El Autor*

3.3.3. ETAPA DE GRABACIÓN

En la etapa de grabación se considera útil una Raspberry Pi3, que permite trabajar con los periféricos de salida como los terminales USB que detalla en la Figura 3.28, donde se utilizan dos puertos USB, de los cuales uno se conecta un convertor USB serial que se utiliza en la recepción de información del Módulo Arduino Mega y en el otro puerto se conecta un HUB USB de 4 puertos, en los cuales se conecta la cámara, la tarjeta de audio y la memoria Micro SD para almacenar la información de un evento.



*Figura. 3. 28 Identificación de puertos USB de la Raspberry Pi
Fuente: El Autor*

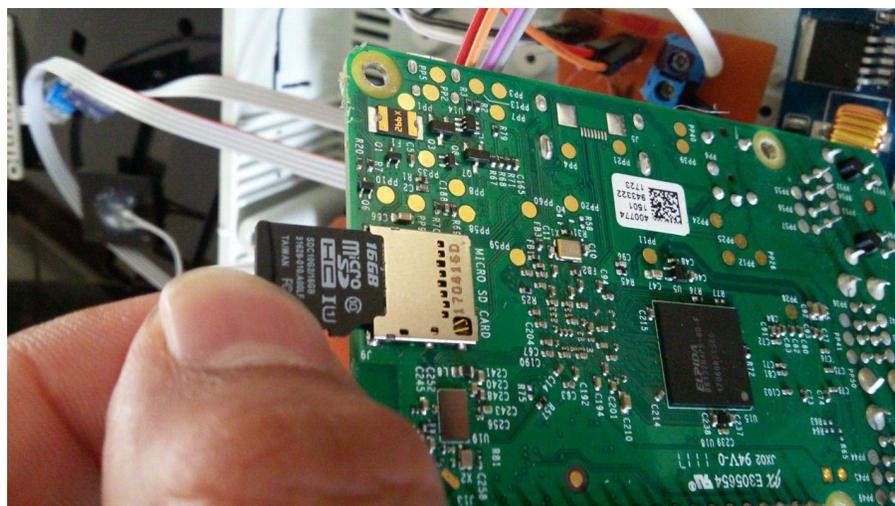
En la etapa de grabación los dispositivos que permiten obtener esta información son la cámara HD de 720P y un micrófono Genius, que tiene mayor ganancia en decibeles para obtener información clara y precisa como se puede observar en la Figura 3.29.



*Figura. 3. 29 Caja de la Cámara HD y Micrófono
Fuente: El Autor*

3.3.4. ETAPA DE ALMACENAMIENTO

Para la etapa de almacenamiento se considera utilizar una memoria Micro SD, en donde guarda la información de la programación realizada, como muestra en la Figura 3.30.



*Figura. 3. 30 Conexión de tarjeta de almacenamiento del Programa
Fuente: El Autor*

También existe otra tarjeta de memoria que se muestra en la Figura 3.31, se utiliza para almacenar información de los eventos ocurridos, en donde contiene información de video, audio y un archivo en formato .TXT que contiene la información de ubicación, velocidad, hora y fecha de cada evento.



*Figura. 3. 31 Conexión de tarjeta de almacenamiento del Programa
Fuente: El Autor*

IMPLEMENTACIÓN DEL DISPOSITIVO EN EL VEHÍCULO

La alimentación del voltaje de la Caja Negra, se realiza desde la batería de manera directa, ya que cuenta con un interruptor que permite el flujo de voltaje al dispositivo.

Ya colocada la fuente de alimentación al interior del vehículo, se realiza la adaptación de la caja del dispositivo en la parte inferior del tablero, de manera que permanezca de forma horizontal por la posición de un acelerómetro-giroscopio. En la Figura 3.32 detalla la forma de colocación del dispositivo.



*Figura. 3. 32 Colocación de la Caja en parte inferior del tablero
Fuente: El Autor*

La adaptación de una caja que sostenga a la cámara y al GPS, se coloca en el parabrisas, como se detalla en la Figura 3.33. Esto se realiza con la finalidad de obtener línea de vista del módulo GPS, y no proporcione datos erróneos.



*Figura. 3. 33 Colocación de la Caja de la cámara en el parabrisas
Fuente: El Autor*

La instalación del sensor de agua se realiza en el guardafango de la llanta delantera, con la finalidad de identificar si la calzada esta mojada, la instalación se demuestra en la Figura 3.34.



*Figura. 3. 34 Colocación del sensor de agua en el guardalado
Fuente: El Autor*

3.4. PRUEBAS DE FUNCIONAMIENTO

Para comprobar que el dispositivo funciona de acuerdo al diseño e implementación del dispositivo, son necesarias las respectivas pruebas de funcionamiento.

Debido a que el dispositivo depende de muchas tareas en secuencia, las pruebas se realizan por etapas. La lista de pruebas que se pueden comprobar en relación a su funcionamiento se describe de la siguiente forma:

3.4.1 ENCENDIDO DE DISPOSITIVOS

En la etapa de encendido se debe verificar que cada uno de los dispositivos reciba la alimentación necesaria para su funcionamiento:

- Raspberry Pi3: Para su alimentación se utiliza una entrada de 12V con un convertor DC-DC a 5V que permite ejecutar los procesos configurados.
- Arduino Mega: En su alimentación depende de un convertor DC-DC de 5V de manera independiente a la Raspberry Pi3, se debe tomar en cuenta que trabajan al mismo voltaje y que sus cargas necesitan de fuentes independientes.
- Acelerómetro-Giroscopio: Depende completamente de la alimentación que se adhiere a la placa PCB del Arduino Mega.
- Módulo GSM SIM 800: Como son dispositivos de la unidad de comunicación su alimentación también depende de la placa PCB que permite la interacción entre el Arduino Mega y el Módulo GSM SIM 800
- Cámara HD: Su única entrada es un puerto USB 2.0 que se conecta directamente en uno de los puertos USB de la Raspberry Pi3 y depende completamente de la misma.

A continuación, se describe las pruebas mencionadas según su funcionamiento en la Tabla 3.7:

Tabla 3. 7 Checklist de encendido de dispositivos errónea

CHECKLIST DE PRUEBAS DE FUNCIONAMIENTO			
ENCENDIDO DE DISPOSITIVOS			
ITEM	PRUEBA DE FUNCIONAMIENTO	FUNCIONAMIENTO	
		SI	NO
1	Encendido de la Raspberry	X	
2	Encendido del Arduino Mega	X	
3	Encendido del Acelerómetro-Giroscópio		X
4	Encendido del GSM SIM 800		X
5	Encendido de la Cámara HD		X

Fuente: El Autor

ITEM 3, 4 y 5. Indica que los elementos no encienden por ende se realiza la siguiente verificación:

- La alimentación considerada para el funcionamiento del proyecto fue muy escasa, por lo que es necesario una alimentación independiente en la unidad de control y grabación.

Una vez adaptado las dos alimentaciones en cada unidad, se realiza las pruebas antes mencionadas en la Tabla 3.8:

Tabla 3. 8 Checklist de encendido de dispositivos corregidos verificada

CHECKLIST DE PRUEBAS DE FUNCIONAMIENTO			
ENCENDIDO DE DISPOSITIVOS			
ITEM	PRUEBA DE FUNCIONAMIENTO	FUNCIONAMIENTO	
		SI	NO
1	Encendido de la Raspberry	x	
2	Encendido del Arduino Mega	x	
3	Encendido del Acelerómetro-Giroscópio	x	
4	Encendido del GSM SIM 800	x	
5	Encendido de la Cámara HD	x	

Fuente: El Autor

3.4.2 ARRANQUE DE LOS SENSORES

Los sensores reciben las señales que permiten ejecutar la programación asignada. En ello se realiza las verificaciones de cada uno, para validar la operatividad del proyecto como detallan las pruebas de la Tabla 3.9:

Tabla 3. 9 Checklist de funcionamiento de los sensores

ARRANQUE DE LOS SENSORES			
ITEM	PRUEBA DE FUNCIONAMIENTO	FUNCIONAMIENTO	
		SI	NO
1	¿Recepta señal el sensor de Agua?	X	
2	¿Recepta señal el sensor de Humo?		X

Fuente: El Autor

ITEM 2. Se identifica que el sensor no se activó al recibir humo, se considera que el sensor no estuvo a la distancia adecuada, para ello se procede de la siguiente forma:

- Se realiza la calibración del sensor de humo con la consideración de activarse al recibir una señal de 650 ppm.

Una vez solventado el inconveniente de esta etapa se puede verificar su prueba en la Tabla 3.10:

Tabla 3. 10 Checklist de ejecución de los sensores verificada

ARRANQUE DE LOS SENSORES			
ITEM	PRUEBA DE FUNCIONAMIENTO	FUNCIONAMIENTO	
		SI	NO
1	¿Recepta señal el sensor de Agua?	X	
2	¿Recepta señal el sensor de Humo?	X	

Fuente: El Autor

3.4.3 ALMACENAMIENTO DE INFORMACIÓN

Esta etapa es una de las más importantes del proyecto, ya que por medio de esta función se puede respaldar cualquier evento y sobre todo necesita de varios elementos para completar el almacenamiento.

En la verificación de esta etapa se obtiene lo descrito en la Tabla 3.11:

Tabla 3. 11 Checklist de ejecución de pruebas de almacenamiento de información errónea

ALMACENAMIENTO DE INFORMACION			
ITEM	PRUEBA DE FUNCIONAMIENTO	FUNCIONAMIENTO	
		SI	NO
1	¿Guarda la información obtenida del Módulo GPS Neo 6M, como también archivos de video y audio si se activa al recibir una señal del Acelerómetro-Giroscopio?		X
2	¿Guarda la información obtenida del Módulo GPS Neo 6M, como también archivos de video y audio si se activa al recibir una señal del sensor de Humo?		X

Fuente: El Autor

ITEM 1 y 2. En esta etapa se analiza el funcionamiento de varios elementos, tomar en cuenta lo siguiente:

- El GPS NEO 6M necesita obtener una buena línea de vista para que pueda proporcionar datos con una distancia aproximada de 4 a 8 metros de error como se puede verificar en la Tabla 3.12.

Tabla 3. 12 Checklist de ejecución de pruebas de precisión del GPS NEO 6M

PRUEBA DE PRECISIÓN DEL GPS						
ITEM	GPS NEO 6M		GPS GARMIN		TIPO DE CLIMA	PRECISIÓN
	LATITUD	LONGITUD	LATITUD	LONGITUD		
1	-0,301324	-78,565724	O 0,3014	S 78,5657	DESPAJADO	4 METROS
2	-0,299338	-78,564324	O 0,2995	S 78,5649	PARCIAMENTE NUBLADO	4 METROS
3	-0,168117	-78,484545	O 0,1674	S 78,4837	NUBLADO	6 METROS
4	-0,197014	-78,492043	O 0,1965	S 78,4906	LLUVIOSO	8 METROS

Fuente: El Autor

Se verifica la línea de vista del GPS para obtener los datos de la Tabla 3.13:

Tabla 3. 13 Checklist de ejecución de pruebas de almacenamiento de información verificada

ALMACENAMIENTO DE INFORMACION			
ITEM	PRUEBA DE FUNCIONAMIENTO	FUNCIONAMIENTO	
		SI	NO
1	¿Guarda la información obtenida del Módulo GPS Neo 6M, como también archivos de video y audio si se activa al recibir una señal del Acelerómetro-Giroscopio?	X	
2	¿Guarda la información obtenida del Módulo GPS Neo 6M, como también archivos de video y audio si se activa al recibir una señal del sensor de Humo?	X	

Fuente: El Autor

3.4.4 ENVÍO DE INFORMACIÓN

El dispositivo permite notificar por medio de un mensaje de texto a un contacto móvil de emergencia, que en el vehículo sucedió un evento.

Para ello las pruebas de funcionamiento se reflejan en la Tabla 3.14

Tabla 3. 14 Checklist de ejecución de pruebas de envío de información errónea

ENVIO DE INFORMACION			
ITEM	PRUEBA DE FUNCIONAMIENTO	FUNCIONAMIENTO	
		SI	NO
1	¿El Módulo GSM SIM 800 envía mensaje de texto al recibir actividad de un Acelerómetro-Giroscopio?		x
2	¿El Módulo GSM SIM 800 envía mensaje de texto al recibir actividad del sensor de Humo?		x

Fuente: El Autor

ITEM 1 y 2 No recibe mensaje de texto porque no hay buena comunicación, donde se verifica lo siguiente:

- La comunicación entre el Módulo Arduino Mega y el Módulo GSM SIM 800 dispone de pines transmisión y recepción, estos tienen que estar conectados de manera inversa para proporcionar una comunicación entre Tx – Rx y Rx –Tx.
- Se comprueba el funcionamiento del Módulo GSM SIM 800 con las siguientes operadoras móviles como se detalla en la Tabla 3.15

Tabla 3. 15 Checklist de pruebas envío de información con las operadoras móviles

PRUEBA DE OPERADOR MÓVIL			
ITEM	OPERADOR	FUNCIONAMIENTO	
		SI	NO
1	MOVISTAR	X	
2	CLARO	X	
3	CNT	X	

Fuente: El Autor

Una vez corregido el inconveniente se realizan las pruebas pertinentes como en la Tabla 3.16:

Tabla 3. 16 Checklist de ejecución de pruebas envío de información verificada

ENVIO DE INFORMACION			
ITEM	PRUEBA DE FUNCIONAMIENTO	FUNCIONAMIENTO	
		SI	NO
1	¿El Módulo GSM SIM 800 envía mensaje de texto al recibir actividad de un Acelerómetro-Giroscopio?	X	
2	¿El Módulo GSM SIM 800 envía mensaje de texto al recibir actividad del sensor de Humo?	X	

Fuente: El Autor

3.4.5 PRUEBAS DE EVENTOS:

Un cambio brusco de velocidad genera un evento que activa todas las funciones de almacenamiento y notificación de la Caja Negra como se refleja en la Tabla 3.17.

Tabla 3. 17 Prueba de Cambio Brusco de Velocidad

PRUEBA DE CAMBIO BRUSCO DE VELOCIDAD									
ITEM	PRUEBA DE FUNCIONAMIENTO	GRABACIÓN		NOTIFICACIÓN SMS		ALARMA SONORA		ALMACENAMIENTO	
		SI	NO	SI	NO	SI	NO	SI	NO
1	Desplazamiento del vehículo a 60 Km/h	X		X		X		x	
2	Desplazamiento del vehículo a 50 Km/h		X		x		x		x
3	Desplazamiento del vehículo a 40 Km/h		X		x		x		x

Fuente: El Autor

ITEM 2 y 3 La prueba realizada a la velocidad de 40 y 50 Km/h no permitió cumplir lo propuesto, por ende, se realiza la siguiente verificación:

- La sensibilidad en el acelerómetro-giroscopio debe receptor un evento a cualquier velocidad.

Una vez calibrada la sensibilidad del acelerómetro-giroscopio se realizan las siguientes pruebas que se refleja en la Tabla 3.18.

Tabla 3. 18 Prueba de cambio brusco de velocidad verificada

PRUEBA DE CAMBIO BRUSCO DE VELOCIDAD									
ITEM	PRUEBA DE FUNCIONAMIENTO	GRABACIÓN		NOTIFICACIÓN SMS		ALARMA SONORA		ALMACENAMIENTO	
		SI	NO	SI	NO	SI	NO	SI	NO
1	Desplazamiento del vehículo a 60 Km/h	X		X		X		x	
2	Desplazamiento del vehículo a 50 Km/h	X		X		X		x	
3	Desplazamiento del vehículo a 40 Km/h	X		X		X		x	

Fuente: El Autor

El desplazamiento de un vehículo en alta velocidad, puede provocar la pérdida de control del automotor y también ocasionar un cambio de posición a 180°, es por eso que existe un giroscopio que permite identificar la posición del vehículo y aprovecha esta señal para notificar a un número de contacto de emergencia como se detalla en la Tabla 3.19.

Tabla 3. 19 Prueba de volcamiento

PRUEBA DE VOLCAMIENTO							
ITEM	PRUEBA DE FUNCIONAMIENTO	GRABACIÓN		NOTIFICACIÓN SMS		ALMACENAMIENTO	
		SI	NO	SI	NO	SI	NO
1	Posición del Dispositivo a 0°		X		X		x
2	Posición del Dispositivo a 45°		X		X		x
3	Posición del Dispositivo a 90°	X		X		X	
4	Posición del Dispositivo a 190°	X		X		X	

Fuente: El Autor

ITEM 1 Y 2 La posición del dispositivo refleja un cambio en la Caja Negra, lo que permite identificar y comprobar que no se ejecuta la etapa de grabación, notificación y almacenamiento mientras su ángulo de inclinación no sea mayor a 45°.

El sensor de humo permite notificar, alertar, grabar y almacenar la información del evento ocurrido, siempre y cuando exista un incendio en el interior del vehículo como se detalla en la Tabla 3.20.

Tabla 3. 20 Pruebas del sensor de Humo

PRUEBA DE FUNCIONAMIENTO DEL SENSOR DE HUMO									
ITEM	PRUEBA DE FUNCIONAMIENTO	GRABACIÓN		NOTIFICACIÓN SMS		ALARMA SONORA		ALMACENAMIENTO	
		SI	NO	SI	NO	SI	NO	SI	NO
1	Humo de Papel Comercio	X		X		X		x	
2	Humo de Tela	X		X		X		x	
3	Humo de Esponja	X		X		X		x	

Fuente: El Autor

3.5. ANÁLISIS DE RESULTADOS

Una vez solucionado los percances, se analiza los resultados obtenidos en el desarrollo del proyecto.

- SENSOR DE AGUA

Las pruebas se realizaron en vías de asfalto, adoquín y una calle con charcos de lodo por el tema de lluvias de la temporada, se considera aproximar la distancia de la vía mojada a 10 metros, así se puede tomar valores precisos, como detalla la Tabla 3.21.

En vías de asfalto y adoquín la concentración de agua en el sensor es del 70%, cantidad suficiente en 10 metros para determinar que en donde que desplaza hay que hacerlo con precaución. La limpieza del sensor es todos los días por la acumulación de polvo. Mientras que en caminos de tierra el sensor permanece sucio, por ende, la limpieza es más frecuente.

Tabla 3. 21 Resultado de pruebas del Sensor de Agua

PRUEBA DE FUNCIONAMIENTO						
ITEM	TIPO DE CALZADA	AGUA	LODO	DISTANCIA DE AGUA O LODO	FUNCIONAMIENTO	
					OK	X
1	ASFALTO	X		10 metros	X	
2	TIERRA		X	10 metros	X	
3	ADOQUIN	X		10 metros	X	

Fuente: El Autor

- SENSOR DE HUMO

Se realiza la simulación con humo de papel, tela y esponja en un recipiente cerca del sensor, para comprobar su sensibilidad. La configuración es por unidades de ppm (partículas por millón). Este elemento está configurado a > 650 ppm

- GSP NEO 6M

La adaptación del GPS se debe realizar estrictamente con línea de vista total, ya que cualquier obstáculo dificulta su comunicación y genera datos erróneos como se refleja en la Tabla 3.22; las pruebas se realizaron cerca de una ventana en el interior del domicilio y en el parabrisas del vehículo una vez ya instalado.

Tabla 3. 22 Resultados de posición del GPS

PRUEBA DE FUNCIONAMIENTO GPS								
ITEM	CLIMA	SIN LÍNEA DE VISTA	CON LÍNEA DE VISTA	DISTANCIA DE ERROR	GRABA AUDIO Y VIDEO	ALMACENA INFORMACIÓN	FUNCIONAMIENTO	
							OK	X
1	NUBLADO		X	6 metros	ok	ok	x	
2	LLOVIOSO		X	8 metros	ok	ok	x	
3	SOLEADO		X	4 metros	ok	ok	x	
4	PARCIALMENTE NUBLADO		X	5 metros	ok	ok	x	
5	SOLEADO	X		SIN DATOS	ok	ok		X
6	NUBLADO	X		SIN DATOS	ok	ok		X

Fuente: El Autor

- GSM SIM 800

Las pruebas realizadas en el envío de SMS fueron exitosas, aunque la información del mensaje puede tardar aproximadamente 1 o 2 minutos, ya que depende de la ubicación en donde el nivel de señal de las operadoras sea superior al 30%. De existir una buena conexión de la red móvil el SMS es inmediato, como se detalla en la Tabla 3.23. Tomar muy en cuenta que el SIM del módulo GSM debe tener saldo para procesar la información del dispositivo.

Tabla 3. 23 Verificación Red Móvil

FUNCIONAMIENTO SMS MÓVIL					
ITEM	TIEMPO DE TARDANZA	POSICIÓN DEL VEHÍCULO	POSICIÓN DEL TF MÓVIL	FUNCIONAMIENTO	
				OK	X
1	00:00:40 seg.	PATIO CON LINEA DE VISTA	INTERIOR DEL DOMICILIO	x	
2	00:00:57 seg.	GARAJE CON CUBIERTA DE LOZA	EXTERIOR DEL DOMICILIO	x	
3	00:00:36 seg.	VEREDA SIN OBSTACULOS	INTERIOR DEL VEHÍCULO	x	

Fuente: El Autor

- RASPBERRY Pi3

Los puertos periféricos del dispositivo están distribuidos de manera permanente, ya que al momento de conectar la cámara o la tarjeta de almacenamiento, la Raspberry los reconocerá como nuevas conexiones y asignará una nueva etiqueta interna en ese puerto, que genera un conflicto al momento de grabar.

CONCLUSIONES

- Se diseñó un dispositivo electrónico entre el Hardware y Software, del Raspberry Pi3 y el Arduino Mega que se comunican a través del conversor serial, donde se verifica una eficaz conexión, que logra controlar la señal de los sensores y actuadores para ejecutar el almacenamiento de información de audio y video en una memoria micro SD.
- Se desarrolló el programa de la placa Arduino Mega y se determinó que el lenguaje C++ es el más común y eficaz como plataforma de programación, para cumplir con la interacción de lectura y comunicación entre los sensores.
- Se implementó una bocina de notificación y un sistema de comunicación basado en tecnología GSM, para el envío de mensajes de texto, donde es más factible utilizar la operadora móvil de Claro por la mayor amplitud de cobertura.
- Se elaboró una placa PCB para los sensores y actuadores que fueron colocados para lograr una conexión desmontable, y al mismo tiempo facilita la sustitución del Arduino Mega en caso de requerirlo.
- Se comprobó que el GPS NEO 6M proporciona datos de la ubicación del vehículo con un factor de error del 1,5%, al mismo tiempo la Raspberry Pi3 genera un archivo .TXT para el almacenamiento de esta información en una memoria micro SD.

RECOMENDACIONES

- La memoria de almacenamiento de los eventos generados contiene un archivo de formato .TXT, por tal motivo no se recomienda cambiar de dispositivo de almacenamiento.
- Implementar en el sistema de comunicación mediante SMS, el envío de coordenadas en donde sucedió un evento para una asistencia inmediata.
- La adquisición de un plan de mensajes de texto, permite mantener una comunicación constante con el contacto de emergencia mediante el sistema de notificación SMS.
- Rediseñar el sistema de cargas de la Caja Negra, permite optimizar el control de la potencia del dispositivo, como también dimensionar los periféricos exactos que se necesitan en el proyecto.
- Para colocar una cámara de visión nocturna se recomienda utilizar la cámara NOIR versión 2 que es compatible con Raspberry Pi3 que proporciona una resolución de 8 Megapíxel con captura de video de 1080p30 para cubrir grabaciones en la noche de mejor calidad.

REFERENCIAS BIBLIOGRAFICAS

- Correo, E. (19 de Junio de 2016). *El Correo*. Recuperado el 01 de Febrero de 2018, de Sociedad: <http://www.elcorreo.com/bizkaia/sociedad/201606/19/caja-negra-coches-permite-20160618215100.html>
- Eduardo, L. (13 de Octubre de 2015). *Hetpro*. Recuperado el 6 de Febrero de 2018, de <http://hetpro-store.com/TUTORIALES/sim900-gsm-shieldarduino>
- Garcia, A. (2015). Caracterización de la mortalidad por accidentes de tránsito en Ecuador. *Revista CienciAmérica*, 22-31.
- Ingeniería MCI Ltda. (12 de 10 de 2004). *Arduino*. Recuperado el 05 de 02 de 2018, de <http://arduino.cl/arduino-mega-2560/>
- Maria, V. (12 de 07 de 2015). Ensayo Sobre Cusas de Accidentes de Transito. *Ensayo Sobre Cusas de Accidentes de Transito*. Ecuador.
- Peñañiel, A. (2016). Mortalidad y Años de Vida Potencialmente perdido pos AT. *Salud Publica de México*.
- Ruiz, A. (2010). *Guía Practica de Sensores*. España: Copyright.
- Sanchez, E. (12 de Junio de 2012). Diseño de un sistema de control domótico basado en la plataforma Arduino. *Diseño de un sistema de control domótico basado en la plataforma Arduino*. Valencia, España.
- Valdés Fernando, P. R. (2007). *Microcontroladores: Fundamentos y aplicaciones con PIC*. España: MARCOMBO.
- Velarde, M. (12 de 07 de 2011). Ensayo Sobre Cusas de Accidentes de Transito. *Ensayo Sobre Cusas de Accidentes de Transito*. Ecuador.

ANEXOS

1. ANEXO CONFIGURACIÓN ARDUINO

```
// Librerías I2C para controlar el MPU6050
// la librería MPU6050.h necesita I2Cdev.h, I2Cdev.h necesita Wire.h
#include "I2Cdev.h"
#include "MPU6050.h"
#include "Wire.h"

// La dirección del MPU6050 puede ser 0x68 o 0x69, dependiendo
// del estado de AD0. Si no se especifica, 0x68 estará implícito
MPU6050 gry_sensor;

// Valores RAW (sin procesar) del acelerómetro y giroscopio en los ejes x,y,z
int ax, ay, az;
int gx, gy, gz;

void setup() {
  Serial.begin(57600); //Inici puerto serial
  Wire.begin();      //Iniciando I2C
  gry_sensor.initialize(); //Iniciando el gry_sensor

  if (gry_sensor.testConnection()) Serial.println("gry_sensor iniciado correctamente");
  else Serial.println("Error al iniciar el gry_sensor");
}
```

```
void loop() {  
    // Leer las aceleraciones y velocidades angulares  
    gry_sensor.getAcceleration(&ax, &ay, &az);  
    gry_sensor.getRotation(&gx, &gy, &gz);  
  
    //Mostrar las lecturas separadas por un [tab]  
    Serial.print("a[x y z] g[x y z]:\t");  
    Serial.print(ax); Serial.print("\t");  
    Serial.print(ay); Serial.print("\t");  
    Serial.print(az); Serial.print("\t");  
    Serial.print(gx); Serial.print("\t");  
    Serial.print(gy); Serial.print("\t");  
    Serial.println(gz);  
  
    delay(100);  
}
```

2. ANEXO PROGRAMACIÓN GPS

GPS

```
// HORARIO UTC
```

```
// https://www.andrews.edu/~tzs/timeconv/timeconvert.php
```

```
#include "TinyGPS.h"
```

```
TinyGPS gps;

static void gps_smartdelay(unsigned long ms);
static void gps_str_date(TinyGPS &gps);
String gps_StrLAT, gps_StrLON, gps_StrVEL, gps_StrDATE = "";

void setup()
{
  Serial.begin(9600);
  Serial2.begin(9600);

void loop()

  gps_StrLAT, gps_StrLON, gps_StrVEL, gps_StrDATE = "";

  float fvel, flat, flon;
  unsigned long age, date, time, chars = 0;
  gps.f_get_position(&flat, &flon, &age);

  gps_StrLAT = String(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flat, 6);
  gps_StrLON = String(flou == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flon, 6);
  gps_StrVEL = String(float(gps.f_speed_kmph()));
  gps_str_date(gps);

  // Serial.println(gps_StrLAT);
```

```
// Serial.println(gps_StrLON);

// Serial.println(gps_StrDATE);

// Serial.println(gps_StrVEL);

String StrGpsTotal = "Latitud: " + gps_StrLAT + " Londitud: " + gps_StrLON + "
Velocidad: " + gps_StrVEL + " kmph Horario UTC: " + gps_StrDATE;

Serial.println(StrGpsTotal);

gps_smartdelay(1000);
}

static void gps_smartdelay(unsigned long ms)
{
    unsigned long start = millis();
    do
    {
        while (Serial2.available())
            gps.encode(Serial2.read());
    } while (millis() - start < ms);
}

static void gps_str_date(TinyGPS &gps)
{
    int year;
    byte month, day, hour, minute, second, hundredths;
    unsigned long age;
```

```

gps.crack_datetime(&year, &month, &day, &hour, &minute, &second, &hundredths, &age);
if (age == TinyGPS::GPS_INVALID_AGE)
    gps_StrDATE = "***** *****";
else
{
    char sz[32];

    sprintf(sz, "%02d/%02d/%02d %02d:%02d:%02d ", month, day, year, hour, minute,
second);

    gps_StrDATE = sz;
}

gps_smartdelay(0);
}

```

/*

DOP = Dilution of precision (Dispersion de la precisión,
o coloquialmente incertidumbre).

Hay varios tipo de DOP: H, V, G, P, T.

VDOP * -- es la incertidumbre en altura (Vertical)

HDOP * -- es la incertidumbre 2D (Horizontal)

PDOP * -- es la incertidumbre 3D

TDOP * -- es la incertidumbre en tiempo

HTDOP * -- es la incertidumbre horizontal y de tiempo

GDOP * -- es la incertidumbre 3D y de tiempo (Geométrica)

Viene determinada por la geometría de los satelites. A diferencia de otros errores que viene dados por la calidad de la señal.

*/

SENSOR LLUVIA HUMO

```
#include <SoftwareSerial.h>
```

```
// pin #2 - Entrada desde el Sensor (cable verde)
```

```
// pin #3 - Salida de desde Arduino (cable blanco)
```

```
SoftwareSerial MySerial(2, 3); // RX, TX
```

```
int PinDeteclluvia = 0;
```

```
int PinDetecHumo = 1;
```

```
int ValAnlglluvia = 0; // variable to store the ValAnlglluviaue read
```

```
int ValAnlgHumo = 0;
```

```
void setup()
```

```
{
```

```
  Serial.begin(9600);
```

```
  Serial.println("Sensores Humo - Lluvia");
```

```
    MySerial.begin(9600);
}

void loop()
{
    ValAnlglluvia = analogRead(PinDetecLluvia); // read the input pin
    ValAnlgHumo = analogRead(PinDetecHumo); // read the input pin

    Serial.println(ValAnlglluvia);
    Serial.println(ValAnlgHumo);

    if (ValAnlgHumo > 650){Serial.println("Humo Detectado");}
    if (ValAnlglluvia < 350){Serial.println("Lluvia Detectada");}

    delay (1000);
}
```

SIM 800

```
char* sim_text;
char* sim_number;
bool sim_error; //to catch the response of sendSms
String sim_buffer;
int sim_timeout;
```

```
void setup()
{
  Serial3.begin(9600); // initialize the library.

  sim_text="OTRO MENSAJE"; //sim_text for the message.
  sim_number="0992728200"; //change to a valid sim_number.
  sim_error = sendSms(sim_number,sim_text);
}

void loop()
{
  //do nothing
}

bool sendSms(char* sim_number,char* sim_text){
  Serial3.print (F("AT+CMGF=1\r")); //set sms to sim_text mode
  sim_buffer=sim_readSerial();
  Serial3.print (F("AT+CMGS=\r")); // command to send sms
  Serial3.print (sim_number);
  Serial3.print(F("\r"));
  sim_buffer=sim_readSerial();
  Serial3.print (sim_text);
  Serial3.print ("\r");
  //change delay 100 to readserial
  sim_buffer=sim_readSerial();
```

```
Serial3.print((char)26);

sim_buffer=sim_readSerial();

//expect CMGS:xxx , where xxx is a sim_number,for the sending sms.

if (((sim_buffer.indexOf("CMGS")) != -1 ) ){

    return true;

}

else {

    return false;

}

}

String sim_readSerial()

{

    sim_timeout=0;

    while (!Serial3.available() && sim_timeout < 12000 )

    {

        delay(13);

        sim_timeout++;

    }

    if (Serial3.available()) {

        return Serial3.readString();

    }

}
```

3. ANEXO 2 INTREGRACION ARDUINO

INTEGRACION ARDUINO

```

#include "TinyGPS.h"

// Librerias I2C para controlar el mpu6050

// la libreria MPU6050.h necesita I2Cdev.h, I2Cdev.h necesita Wire.h

#include "I2Cdev.h"

#include "MPU6050.h"

#include "Wire.h"

#define SrInfArduino Serial

#define SrRaspberry Serial1

#define SrModGPS Serial2

#define SrModGSM Serial3

boolean SendRaspberry = false;

boolean BoolINii_msg = false;

// La dirección del MPU6050 puede ser 0x68 o 0x69, dependiendo
// del estado de AD0. Si no se especifica, 0x68 estará implícito

MPU6050 gry_sensor;

// Valores RAW (sin procesar) del acelerometro y giroscopio en los ejes x,y,z

int ax, ay, az;

int gx, gy, gz;

//-----SIM 800 -----

char* sim_text;

char* sim_number;

```

```

bool sim_error; //to catch the response of sendSms

String sim_buffer;

int sim_timeout;

//----- SENSORES HUMO LLUVIA ----

int PinDetecluvia = 0;

int PinDetecHumo = 1;

int ValAnlglluvia = 0;      // variable to store the ValAnlglluviaue read

int ValAnlgHumo = 0;

const byte Chicharra = 3;

const byte IntAclGiros = 7;

//-----MODULO GPS -----

TinyGPS gps;

static void gps_smartdelay(unsigned long ms);

static void gps_str_date(TinyGPS &gps);

String gps_StrLAT, gps_StrLON, gps_StrVEL, gps_StrDATE = "";

void VerificarGPS (void);

//-----

void LeerAclrGiroscopio (void);

void DetecluviaHumo (void);

void sim_EnviarMensaje (void);

//-----

void setup()

{

    SrInfArduino.begin(9600);

```

```
SrRaspberry.begin(9600);
SrModGPS.begin(9600);
SrModGSM.begin(9600);
Wire.begin();    //Iniciando I2C
pinMode(Chicharra, OUTPUT);
pinMode(IntAclGiros, INPUT);
    digitalWrite(Chicharra, HIGH);
delay(500);
digitalWrite(Chicharra, LOW);

SrInfArduino.println("INTEGRACION SISTEMA CAJA NEGRA");
// SrRaspberry.println("Primera trama");

gry_sensor.initialize(); //Iniciando el gry_sensor
if (gry_sensor.testConnection()) SrInfArduino.println("gry_sensor iniciado correctamente");
else SrInfArduino.println("Error al iniciar el gry_sensor");
SendRaspberry = false;
VerificarGPS();delay(100);
VerificarGPS();delay(100);
VerificarGPS();delay(100);
SendRaspberry = true;
BoolINii_msg = true;
sim_EnviarMensaje();
BoolINii_msg = false;
}
```

```

void loop()
{
  DetecLluviaHumo();
  LeerAclrGiroscopio();
}

void LeerAclrGiroscopio (void)
{
  // Leer las aceleraciones y velocidades angulares
  gry_sensor.getAcceleration(&ax, &ay, &az);
  gry_sensor.getRotation(&gx, &gy, &gz);

  /*
  //Mostrar las lecturas separadas por un [tab]
  SrInfArduino.print("a[x y z] g[x y z]:\t");
  SrInfArduino.print(ax); SrInfArduino.print("\t");
  SrInfArduino.print(ay); SrInfArduino.print("\t");
  SrInfArduino.print(az); SrInfArduino.print("\t");
  SrInfArduino.print(gx); SrInfArduino.print("\t");
  SrInfArduino.print(gy); SrInfArduino.print("\t");
  SrInfArduino.println(gz);
  */

  if ((ax > 22000) || (ay > 22000)) { MensajeEmergenciaGrabar(2); }
  if ((gx > 7000) || (gx < -7000)) { MensajeEmergenciaGrabar(3); }
  if ((gy > 7000) || (gy < -7000)) { MensajeEmergenciaGrabar(3); }
}

```

```

}

void VerificarGPS (void)
{
    gps_StrLAT, gps_StrLON, gps_StrVEL, gps_StrDATE = "";

    float fvel, flat, flon;

    unsigned long age, date, time, chars = 0;

    gps.f_get_position(&flat, &flon, &age);

    gps_StrLAT = String(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flat, 6);
    gps_StrLON = String(flou == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flon, 6);
    gps_StrVEL = String(float(gps.f_speed_kmph()));

    gps_str_date(gps);

    // SrInfArduino.println(gps_StrLAT);

    // SrInfArduino.println(gps_StrLON);

    // SrInfArduino.println(gps_StrDATE);

    // SrInfArduino.println(gps_StrVEL);

    String StrGpsTotal = "Latitud: " + gps_StrLAT + " Longitud: " + gps_StrLON + "
Velocidad: " + gps_StrVEL + " kmph Horario UTC: " + gps_StrDATE;

    SrInfArduino.println(StrGpsTotal);

    if (SendRaspberry){SrRaspberry.println(StrGpsTotal);}

    gps_smartdelay(1000);
}

static void gps_smartdelay(unsigned long ms)
{
    unsigned long start = millis();

    do

```

```

{
    while (SrModGPS.available())
        gps.encode(SrModGPS.read());
    } while (millis() - start < ms);
}

static void gps_str_date(TinyGPS &gps)
{
    int year;
    byte month, day, hour, minute, second, hundredths;
    unsigned long age;
    gps.crack_datetime(&year, &month, &day, &hour, &minute, &second, &hundredths, &age);
    if (age == TinyGPS::GPS_INVALID_AGE)
        gps_StrDATE = "*****";
    else
    {
        char sz[32];

        sprintf(sz, "%02d/%02d/%02d %02d:%02d:%02d ", month, day, year, hour, minute,
second);

        gps_StrDATE = sz;
    }
    gps.smartdelay(0);
}

void MensajeEmergenciaGrabar (byte Evento)
{

```

```

sim_EnviarMensaje();

VerificarGPS();

delay (100);

if (Evento == 1){SrRaspberry.println("HUMO");SrInfArduino.println("HUMO");}

if (Evento == 2){SrRaspberry.println("CHOQUE");SrInfArduino.println("CHOQUE");}

if (Evento ==
3){SrRaspberry.println("VOLCAMIENTO");SrInfArduino.println("VOLCAMIENTO");}

delay(2000);

}

/*

```

DOP = Dilution of precission (Dispersion de la precisión,
o coloquialmente incertidumbre).

Hay varios tipo de DOP: H, V, G, P, T.

VDOP * -- es la incertidumbre en altura (Vertical)

HDOP * -- es la incertidumbre 2D (Horizontal)

PDOP * -- es la incertidumbre 3D

TDOP * -- es la incertidumbre en tiempo

HTDOP * -- es la incertidumbre horizontal y de tiempo

GDOP * -- es la incertidumbre 3D y de tiempo (Geométrica)

Viene determinada por la geometría de los satelites. A diferencia de otros
errores que viene dados por la calidad de la señal.

```
*/
```

```
void DeteclluviaHumo (void)
```

```
{  
  ValAnlglluvia = analogRead(PinDeteclluvia); // read the input pin  
  ValAnlgHumo = analogRead(PinDetecHumo); // read the input pin  
  //SrInfArduino.println(ValAnlglluvia);  
  //SrInfArduino.println(ValAnlgHumo);  
  
  if (ValAnlgHumo > 650)  
  {  
    SrInfArduino.println("Humo Detectado");  
    MensajeEmergenciaGrabar(1);  
  }  
  if (ValAnlglluvia < 350)  
  {  
    //SrInfArduino.println("Lluvia Detectada");  
    digitalWrite(Chicharra, HIGH);  
  }  
  else  
  {  
    digitalWrite(Chicharra, LOW);  
  }  
}  
  
void sim_EnviarMensaje (void)  
{  
  if(BoolINii_msg)
```

```

{
  sim_text="Iniciando Modulo GSM"; //sim_text for the message.
}
else
{
  sim_text="Peligro Automovil - Verificar Estado"; //sim_text for the message.
}

sim_number="0992728200"; //change to a valid sim_number.
SrInfArduino.println("enviando mensaje SIM 800");
sim_error = sendSms(sim_number,sim_text);
}

```

```

bool sendSms(char* sim_number,char* sim_text){
  SrModGSM.print (F("AT+CMGF=1\r")); //set sms to sim_text mode
  sim_buffer=sim_readSrInfArduino();
  SrModGSM.print (F("AT+CMGS=\r")); // command to send sms
  SrModGSM.print (sim_number);
  SrModGSM.print(F("\r"));
  sim_buffer=sim_readSrInfArduino();
  SrModGSM.print (sim_text);
  SrModGSM.print ("\r");
  //change delay 100 to readSrInfArduino
  sim_buffer=sim_readSrInfArduino();
  SrModGSM.print((char)26);
  sim_buffer=sim_readSrInfArduino();
}

```

```

//expect CMGS:xxx , where xxx is a sim_number,for the sending sms.
if (((sim_buffer.indexOf("CMGS")) != -1 )){
    return true;
}
else {
    return false;
}
}

```

```

String sim_readSrInfArduino()
{
    sim_timeout=0;
    while (!SrModGSM.available() && sim_timeout < 12000 )
    {
        delay(13);
        sim_timeout++;
    }
    if (SrModGSM.available()) {
        return SrModGSM.readString();
    }
}

```

4. ANEXO CONFIGURACION RASPBERRY

RASPBERRY

CAPTURA AUDIO

```
import os, sys

import subprocess

#os.system("arecord --device=hw:1,0 --format S16_LE --rate 44100 -c1
/media/pi/AUDIOVIDEO/RESPALDOS/myAudio.wav")

import subprocess

#LEO ARCHIVO QUE ASIGNA NUMERO DE NOMBRE

NameFile = open("/home/pi/Desktop/RaspArd/NameFile.txt", "r")

NumeroGuardado = NameFile.read()

NameFile.close()

if NumeroGuardado == "":

    NumeroGuardado =0

print "iniciando"

record = "arecord --device=hw:1,0 --format S16_LE --rate 44100 -c1
/media/pi/AUDIOVIDEO/RESPALDOS/Audio"+str(NumeroGuardado)+".wav"

p = subprocess.Popen(record, shell=True)

print "saliendo"

# valor1 = os.system("arecord --device=hw:1,0 --format S16_LE --rate 44100 -c1
/media/pi/AUDIOVIDEO/RESPALDOS/myAudio.wav")

# print("Resultado:", valor1)
```

CAPTURA VIDEO

```

# -*- coding: utf-8 -*-

#!/usr/bin/env Python2

import cv2

import numpy as np

import os

import imutils

Video_Camara = 0

WidthVideo = 300

ValorScaleFactor = 1.2

# Define the codec and create VideoWriter object

fourcc = cv2.VideoWriter_fourcc(*'XVID')

out = cv2.VideoWriter('/media/pi/AUDIOVIDEO/RESPALDOS/Evento.avi',fourcc, 20.0,
(640,480))

#####
#####

def main():

    capWebcam = cv2.VideoCapture(Video_Camara)    # declare a VideoCapture object and
associate to webcam, 0 => use 1st webcam

    if capWebcam.isOpened() == False:          # check if VideoCapture object was associated
to webcam successfully

        print "error: capWebcam not accessed successfully\n\n"    # if not, print error message to
std out

        os.system("pause")                                # pause until user presses a key so user can
see error message

        return                                            # and exit function (which exits program)

    while cv2.waitKey(1) != 27 and capWebcam.isOpened():    # until the Esc key is
pressed or webcam connection is lost

```

```

    blnFrameReadSuccessfully, imgOriginal = capWebcam.read()      # read next frame

    if not blnFrameReadSuccessfully or imgOriginal is None:    # if frame was not read
successfully
        print "error: frame not read from webcam\n"          # print error message to std out
        os.system("pause")                                    # pause until user presses a key so user can
see error message
        break                                                # exit while loop (which exits program)

    out.write(imgOriginal)

    # resize the frame and convert it to grayscale
    imgOriginal = imutils.resize(imgOriginal, width = WidthVideo)
    cv2.imshow("imgOriginal", imgOriginal)    # show windows
    # Flip the image horizontally
    # flipped = cv2.flip(imgOriginal, 1)
    # cv2.imshow("Flipped Horizontally", flipped)

    cv2.destroyAllWindows()    # remove windows from memory

    return

#####
#####

if __name__ == "__main__":
    main()

```

5. ANEXO CONFIGURACION DE COMUNICACION ARDUINO

COMUNICACION SRDUINO

```
# -*- coding: utf-8 -*-
```

```
#!/usr/bin/python2
```

```
import runpy, sys

import serial

import time

import os

import commands

arduino=serial.Serial('/dev/ttyUSB0',baudrate=9600, timeout = 3.0)

BufferIn_Arduino=""

TramaComparar = "

while True:

while True:

    # leemos hasta que encontremos el final de linea

    BufferIn_Arduino = arduino.readline()

    # Mostramos el valor leído y eliminamos el salto de linea del final

    if len(BufferIn_Arduino) > 0:

        buffer_in = open("/media/pi/AUDIOVIDEO/RESPALDOS/GpsInfo.txt", "a")

        buffer_in.write(BufferIn_Arduino+ '\n')

        buffer_in.close()

        # print "Trama Leida: " + BufferIn_Arduino.rstrip('\n')

        TramaComparar = BufferIn_Arduino

        print TramaComparar.rstrip('\n')

        BufferIn_Arduino = ""

    if TramaComparar.find('HUMO') >= 0:

        print "HUMO - Iniciando grabacion de video y audio"

        execfile('IntSerial.py')

        print "CONTINUANDO SERIAL"
```

```
if TramaComparar.find('CHOQUE') >= 0:
    print "CHOQUE - Iniciando grabacion de video y audio"
    execfile('IntSerial.py')
    print "CONTINUANDO SERIAL"
        if TramaComparar.find('VOLCAMIENTO') >= 0:
            print "VOLCAMIENTO - Iniciando grabacion de video y audio"
            execfile('IntSerial.py')
            print "CONTINUANDO SERIAL"

if TramaComparar.find('stop') >= 0:
    print "Deteniendo grabacion de video y audio"

TramaComparar = "

arduino.close()
```
