



**UNIVERSIDAD TÉCNICA DE AMBATO**

**FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E  
INDUSTRIAL**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y  
COMUNICACIONES**

**Tema:**

---

**“SISTEMA ELECTRÓNICO PARA EL CONTROL DE SEGURIDAD FÍSICA Y  
CONDUCCIÓN ASISTIDA DE UNA MOTOCICLETA”**

---

Trabajo de Graduación. Modalidad: Proyecto de Investigación, presentado previo la obtención del título de Ingeniero en Electrónica y Comunicaciones.

**SUBLÍNEA DE INVESTIGACIÓN:** Sistemas Embebidos

**AUTOR:** Pérez Nasser Jovann

**TUTOR:** Ing. Geovanni Brito, Mg.

Ambato – Ecuador

Julio 2015

## **APROBACIÓN DEL TUTOR**

En mi calidad de tutor del trabajo de investigación sobre el tema “Sistema Electrónico para el control de Seguridad Física y Conducción Asistida de una Motocicleta”, del señor: Jovann Pérez Nasser, estudiante de la carrera de Ingeniería en Electrónica y en Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el numeral 7.2 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de la Facultades de la Universidad Técnica de Ambato.

Ambato, Julio de 2015

EL TUTOR

---

Ing. Geovanni Danilo Brito Moncayo, Mg.

## **AUTORÍA**

El presente trabajo de investigación Titulado **“SISTEMA ELECTRÓNICO PARA EL CONTROL DE SEGURIDAD FÍSICA Y CONDUCCIÓN ASISTIDA DE UNA MOTOCICLETA”**. Es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, Julio de 2015

---

Jovann Pérez Nasser

CC: 1804823746

## **DERECHOS DE AUTOR**

Autorizo a la Universidad Técnica de Ambato para que haga uso de este Trabajo de titulación como un documento disponible para lectura, consulta y procesos de investigación.

Cedo los derechos de mi trabajo de Titulación con fines de difusión pública, además autorizo su reproducción dentro de las regulaciones de la Universidad.

Ambato, Julio de 2015.

---

Jovann Pérez Nasser

CC: 1804823746

## **APROBACIÓN DEL TRIBUNAL DE GRADO**

La comisión calificadora del presente trabajo conformada por los señores docentes Ing. Julio Cuji Rodríguez e Ing. Marco Jurado Lozada, revisó y aprobó el informe final del trabajo de graduación titulado Sistema Electrónico para el control de Seguridad Física y Conducción Asistida de una Motocicleta, presentado por el señor Jovann Pérez Nasser de acuerdo al Art. 17 del Reglamento de Graduación para Obtener el Título Terminal de Tercer Nivel de la Universidad Técnica de Ambato.

---

Ing. Vicente Morales, Mg.

**PRESIDENTE DEL TRIBUNAL**

---

Ing. Julio Cuji Rodríguez, Mg

**DOCENTE CALIFICADOR**

---

Ing. Marco Jurado Lozada, Mg

**DOCENTE CALIFICADOR**

## **DEDICATORIA**

A mi madre. A ella, la silenciosa y magnífica artífice detrás de cada uno de los pasos que he dado por este camino de la vida. A mi hermano. Y a mis adoradas mascotas: Wishbone (+), Balú y Matías.

Jovann Pérez

## AGRADECIMIENTO

*“Daría todo lo que sé por la mitad de lo que ignoro” – René Descartes.* Las personas, al igual que las estrellas, debemos brillar por nuestro conocimiento. En este pequeño paso hacia la búsqueda de la luz de la verdad, quiero agradecer a mi tutor y a mis calificadores, figuras insignes de mi facultad. Agradecerles por sus enseñanzas y su paciencia. A mi ahora colega, Ing. Santiago Villacís, quien siempre supo extender una mano amiga en mis momentos de duda, y quien es uno de los mejores profesionales en el campo de la electrónica que he tenido el gusto de conocer. A Max, mi gran amigo. A Jorge, David, Andrés, Luis, Beto, André, Jorge David. A toda la pandilla de desquiciados y desquiciadas con quienes he tenido el extraño gusto de caminar en estos años.

Jovann Pérez

## ÍNDICE

<b>APROBACIÓN DEL TUTOR</b>	<b>i</b>
<b>AUTORÍA</b>	<b>ii</b>
<b>DERECHOS DE AUTOR</b>	<b>iii</b>
<b>APROBACIÓN DE LA COMISIÓN CALIFICADORA</b>	<b>iv</b>
<b>Resumen</b>	<b>xv</b>
<b>Abstract</b>	<b>xvi</b>
<b>Glosario de Términos</b>	<b>xvii</b>
<b>Introducción</b>	<b>xx</b>
<b>CAPÍTULO I</b> .....	<b>1</b>
<b>EL PROBLEMA</b> .....	<b>1</b>
1.1.    TEMA .....	1
1.2.    PLANTEAMIENTO DEL PROBLEMA .....	1
1.2.1.    CONTEXTUALIZACIÓN .....	1
1.3.    DELIMITACIÓN DE CONTENIDOS .....	2
1.4.    JUSTIFICACIÓN .....	3
1.5.    OBJETIVOS .....	4
1.5.1.    OBJETIVO GENERAL .....	4
1.5.2.    OBJETIVOS ESPECÍFICOS .....	4
<b>CAPÍTULO 2</b> .....	<b>5</b>
<b>MARCO TEÓRICO</b> .....	<b>5</b>
2.1.    ANTECEDENTES INVESTIGATIVOS .....	5
2.2.    FUNDAMENTACIÓN TEÓRICA .....	5
2.2.1.    SISTEMAS ELECTRÓNICOS DE LAS MOTOCICLETAS .....	5
2.2.1.1.    EVOLUCIÓN DE LOS SISTEMAS ELECTRÓNICOS EN LA INDUSTRIA AUTOMOTRIZ .....	8



2.2.1.2.	SISTEMA ELECTRÓNICO DE LAS MOTOCICLETAS MOTOR 1 ...	9
2.2.2.	SENSORES EN LA INDUSTRIA AUTOMOTRIZ .....	10
2.2.2.1.	SENSOR DE EFECTO HALL.....	13
2.2.3.	HARDWARE LIBRE .....	16
2.2.3.1.	ARDUINO.....	18
2.2.3.1.1.	LENGUAJE DE PROGRAMACIÓN ARDUINO.....	19
2.2.3.1.2.	AVR VS PIC.....	20
2.2.4.	MÓDULOS GSM GPS .....	21
2.2.4.1.	PROGRAMACIÓN DE LOS MÓDULOS GSM GPS.....	22
2.2.5.	SISTEMAS DE IDENTIFICACIÓN BIOMÉTRICOS.....	24
2.2.5.1.	PROCESO DE RESOLUCIÓN DE IDENTIDAD .....	24
2.2.5.2.	TIPOS DE SISTEMAS BIOMÉTRICOS .....	25
2.2.5.3.	TECNOLOGÍAS BIOMÉTRICAS ESTÁTICAS .....	26
2.2.5.3.1.	RECONOCIMIENTO DE VOZ.....	27
2.2.5.3.1.1.	PROCESO DE RECONOCIMIENTO DE VOZ.....	28
2.2.6.	INTERFACES DE SALIDA.....	29
2.2.6.1.	TIPOS DE INTERFACES DE SALIDA .....	29
<b>CAPÍTULO 3</b>	<b>.....</b>	<b>32</b>
<b>METODOLOGÍA</b>	<b>.....</b>	<b>32</b>
3.1.	MODALIDAD BÁSICA DE LA INVESTIGACIÓN.....	32
3.2.	RECOLECCIÓN DE INFORMACIÓN .....	33
3.2.1.	PLAN DE RECOLECCIÓN DE INFORMACIÓN .....	33
3.3.	PROCESAMIENTO Y ANÁLISIS DE LA INFORMACIÓN .....	33
3.4.	DESARROLLO DEL PROYECTO .....	33
<b>CAPÍTULO 4</b>	<b>.....</b>	<b>34</b>
<b>LA PROPUESTA</b>	<b>.....</b>	<b>34</b>
4.1.	ANTECEDENTES DE LA PROPUESTA .....	34
4.2.	DIAGRAMA DE BLOQUES DEL SISTEMA .....	35
4.3.	CRITERIOS DE SELECCIÓN DE EQUIPOS.....	37
4.4.	SELECCIÓN DEL SISTEMA MICROCONTROLADOR.....	37

4.4.1.	EVALUACIÓN DE LOS SISTEMAS MICROCONTROLADOS .....	39
4.4.2.	ARDUINO MEGA 2560 R3 .....	41
4.4.2.1.	CARACTERÍSTICAS ELÉCTRICAS ARDUINO MEGA 2560 R3.....	41
4.4.3.	ARDUINO UNO SMD R3 .....	42
4.4.3.1.	CARACTERÍSTICA ELÉCTRICAS ARDUINO UNO SMD R3 .....	43
4.5.	SELECCIÓN DEL MÓDULO GSM/GPS .....	44
4.5.1.	EVALUACIÓN DE LOS MÓDULOS GSM/GPS.....	45
4.6.	SIM908 KITS UNO .....	46
4.6.1.	PROGRAMACIÓN SIMCOM 908 .....	48
4.6.1.1.	COMANDOS HAYES PARA SIMCOM 908 .....	48
4.6.1.2.	LIBRERÍAS COMPATIBLES CON SIMCOM 908 .....	48
4.7.	SELECCIÓN DEL SISTEMA BIOMÉTRICO.....	48
4.7.1.	EVALUACIÓN DE LOS SISTEMAS BIOMÉTRICOS .....	50
4.8.	EASY VR SHIELD 2.0.....	51
4.8.1.	CARACTERÍSTICAS ELÉCTRICAS EASY VR 2.0 .....	52
4.8.2.	COMUNICACIÓN SERIAL DE EASY VR 2.0.....	53
4.8.3.	CONFIGURACIONES FÍSICAS EASY VR 2.0 .....	53
4.8.4.	CONSIDERACIONES ADICIONALES EASY VR 2.0.....	55
4.9.	SELECCIÓN DE LA INTERFAZ DE SALIDA.....	55
4.9.1.	EVALUACIÓN DE LAS INTERFACES DE SALIDA .....	56
4.10.	NOKIA 5110 LCD DISPLAY .....	57
4.10.1.	DISPOSICIÓN DE PINES NOKIA 5110 LCD DISPLAY .....	58
4.11.	BLOQUE DE SEGURIDAD FÍSICA .....	59
4.11.1.	CIRCUITO DE ENCENDIDO POR CDI.....	60
4.11.2.	DISEÑO DEL BLOQUE DE SEGURIDAD FÍSICA .....	62
4.11.3.	COMANDOS DE VOZ.....	63
4.11.4.	COMANDOS DE RED .....	64
4.11.4.1.	COMANDO USER .....	64
4.11.4.2.	COMANDOS ROOT .....	65
4.11.4.3.	FORMATO DE RESPUESTA MEDIANTE SMS .....	66

4.11.5.	DIAGRAMA DE FLUJO DEL BLOQUE DE CONTROL DE SEGURIDAD FÍSICA .....	66
4.12.	BLOQUE DE CONDUCCIÓN ASISTIDA .....	68
4.12.1.	DISEÑO DEL BLOQUE DE CONDUCCIÓN ASISTIDA .....	68
4.12.2.	DIAGRAMA DE FLUJO DEL BLOQUE DE CONDUCCIÓN ASISTIDA.....	71
4.13.	IMPLEMENTACIÓN DEL PROTOTIPO .....	72
4.13.1.	ALIMENTACIÓN DEL SISTEMA.....	72
4.13.2.	PLACA DE ACTUADORES.....	74
4.13.2.1.	SIMULACIONES .....	74
4.13.2.2.	DISEÑO DE LA PLACA DE ACTUADORES .....	75
4.13.3.	CIRCUITO FINAL .....	77
4.13.4.	ADECUACIONES DEL CABLEADO DE LA MOTOCICLETA .....	78
4.13.5.	MONTAJE DE LA CAJA E INSTALACIÓN .....	84
4.14.	PRUEBAS DEL PROTOTIPO .....	85
4.14.1.	PRUEBAS DEL SISTEMA ELECTRÓNICO DE CONTROL DE SEGURIDAD FÍSICA .....	86
4.14.2.	PRUEBAS DEL SISTEMA ELECTRÓNICO DE CONDUCCIÓN ASISTIDA .....	87
4.15.	PRESUPUESTO DEL PROYECTO .....	90
<b>CAPÍTULO V.....</b>	<b>92</b>	
<b>CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>92</b>	
5.1	CONCLUSIONES .....	92
5.2	RECOMENDACIONES.....	92

## ÍNDICE DE TABLAS

Tabla 2.1: Resumen de características Dakar 200cc.....	9
Tabla 2.2: Clases de factores externos que debe resistir un sensor automotriz .....	11
Tabla 2.3: Microcontroladores utilizados por las placas Arduino .....	18
Tabla 2.4: Clasificación de la familia AVR fabricada por ATMEL.....	19
Tabla 2.5: Listado de comandos HAYES principales .....	23
Tabla 2.6: Tabla de resumen de Tecnologías Estáticas .....	26
Tabla 2.7: Ventajas y Desventajas del reconocimiento de voz.....	27
Tabla 4.1: Características de los sistemas microcontrolados preseleccionados .....	38
Tabla 4.2: Calificación de parámetros de selección para el sistema microcontrolador ..	40
Tabla 4.3: Características Eléctricas Arduino Mega 2560 R3.....	42
Tabla 4.4: Características Eléctricas Arduino Uno SMD R3 .....	43
Tabla 4.5: Características de módulos GSM/GSPS preseleccionados .....	44
Tabla 4.6: Calificación de parámetros de selección para el módulo GSM/GPS .....	46
Tabla 4.7: Resumen General SIMCOM908 .....	47
Tabla 4.8: Estados del LED indicador SIMCOM 908.....	47
Tabla 4.9: Características de los módulos biométricos preseleccionados .....	49
Tabla 4.10: Calificación de parámetros de selección para el módulo biométrico .....	51
Tabla 4.11: Características eléctricas EasyVR Shield 2.0 .....	52
Tabla 4.12: Características piezoeléctricas EasyVR Shield 2.0 .....	53
Tabla 4.13: Modos de operación de EasyVR Shield .....	54
Tabla 4.14: Características de los displays preseleccionados.....	56
Tabla 4.15: Calificación de parámetros de selección para interfaces de salida .....	57
Tabla 4.17: Listado de comandos de Voz.....	64
Tabla 4.18: Listado de comandos USER .....	65
Tabla 4.19: Listado de comandos ROOT .....	65
Tabla 4.20: Umbrales de velocidad para cambios de marcha de una motocicleta .....	68
Tabla 4.21: Tensiones y corrientes de operación de los elementos del sistema .....	72

Tabla 4.22: Comparativa de consumo de combustible en una motocicleta sin y con sistema electrónico de conducción asistida .....	88
Tabla 4.23: Presupuesto del proyecto .....	90

## ÍNDICE DE FIGURAS

Figura 2.1: Diagrama de conexiones del ECU de una motocicleta Aprillia.....	6
Figura 2.2: Exigencias y medidas de desarrollo de los sensores automotrices.....	11
Figura 2.3: Niveles de integración de un sensor automotriz.....	12
Figura 2.4: Explicación gráfica del Efecto Hall .....	13
Figura 2.5: Estructura de un sensor de efecto Hall .....	15
Figura 2.6: Consola Uzebox .....	16
Figura 2.7: Módulo GSM/GPS para Arduino .....	21
Figura 2.8: Procesos biométricos de reconocimiento y verificación .....	25
Figura 2.9: Diagrama de Bloques de un sistema de reconocimiento de voz .....	28
Figura 2.10: Display LCD de 2x16.....	29
Figura 4.1: Diagrama de bloques del sistema .....	36
Figura 4.2: Arduino Mega 2560 R3.....	41
Figura 4.3: Arduino Uno SMD R3 .....	43
Figura 4.4: Módulo SIM 908, SIMCOM.....	46
Figura 4.5: EasyVR shield 2.0 para Arduino.....	52
Figura 4.6: Detalle físico EasyVR Shield 2.0.....	54
Figura 4.7: Aspecto físico Nokia LCD5110 .....	58
Figura 4.8: Controles de la mano derecha de una motocicleta .....	59
Figura 4.9: Circuito de encendido mediante descarga por condensador (CDI).....	60
Figura 4.10: Descarga del condensador a través del transformador .....	61
Figura 4.11: Circuito de ignición de una motocicleta.....	62
Figura 4.12: Circuito de pruebas bloque de control de seguridad .....	63
Figura 4.13: Diagrama de flujo para el bloque de seguridad.....	67
Figura 4.14: Conjunto velocímetro – tacómetro de una motocicleta.....	69
Figura 4.15: Sensor de velocidad de Efecto Hall de una motocicleta .....	70
Figura 4.16: Circuito de pruebas bloque de conducción asistida .....	70
Figura 4.17: Diagrama de flujo para el sistema de conducción asistida.....	71
Figura 4.18: Simulación del circuito de la placa de actuadores.....	74

Figura 4.19: Diagrama esquemático circuito de actuación.....	75
Figura 4.20: Diagrama pictórico circuito de actuación.....	76
Figura 4.21: Pistas adheridas a la baqueta .....	76
Figura 4.22: Placa de actuadores ensamblada .....	77
Figura 4.23: Diagrama de bloques del prototipo .....	77
Figura 4.24: Controles de la mano derecha de una motocicleta .....	79
Figura 4.25: Ubicación de la batería en la motocicleta de prueba.....	79
Figura 4.26: Tensión nominal de la batería con el motor apagado.....	80
Figura 4.27: Tensión nominal de la batería con el motor encendido.....	80
Figura 4.28: Identificación del cableado de la columna de ignición .....	81
Figura 4.29: Identificación de los pares del switch de emergencia y del pulsador de arranque .....	81
Figura 4.30: Desmontaje del interruptor de la mano derecha.....	82
Figura 4.31: Cortocircuito del switch de emergencia .....	82
Figura 4.32: Desmontaje de los cables del pulsador de arranque.....	83
Figura 4.33: Corriente de ignición de encendido.....	83
Figura 4.34: Pruebas de ignición mediante relés .....	83
Figura 4.35: Ensamble de la caja con perforaciones para cableado y ventilación.....	84
Figura 4.36: Disposición de las ranuras de ventilación de la caja .....	84
Figura 4.37: Caja del prototipo .....	84
Figura 4.38: Prototipo instalado en la caja .....	85
Figura 4.39: Montaje del prototipo en la motocicleta.....	85
Figura 4.40: Tiempos de respuesta estimados del prototipo.....	87
Figura 4.41: Comparativa de consumo de combustible en una motocicleta sin y con sistema electrónico de conducción asistida .....	89

## RESUMEN

El presente proyecto de investigación detalla la implementación de un sistema prototipo para el bloqueo de encendido, rastreo y conducción asistida de una motocicleta. Se ha procedido a digitalizar los controles del encendido eléctrico, así también como monitorear la velocidad del vehículo. Se ha optado por la utilización de Arduino como plataforma de desarrollo para actuar como unidad microcontroladora del sistema embebido. El prototipo cuenta con dos bloques: uno de seguridad y otro de conducción asistida. El bloque de seguridad utiliza un EasyVR 2.0, un robusto módulo de reconocimiento de voz dependiente de usuario. Se incluye además un módulo GSM/GPS basado en el circuito integrado SIM908, que permite recuperar la ubicación de la motocicleta mediante coordenadas o apagar remotamente el motor en caso de posible hurto o sustracción del vehículo. El bloque de conducción asistida monitorea constantemente la velocidad de la motocicleta para sugerir una marcha acorde a una zona de bajas revoluciones del motor, en pos de limitar el consumo de carburante. La utilización del sistema prototipo ha incrementado de manera notable la seguridad de la motocicleta frente a escenarios delictivos, así también como ha permitido economizar en lo que a consumo de combustible se refiere. Estas características brindan un alto valor agregado a un vehículo que ha visto incrementado su mercado, debido a su creciente popularidad como medio de transporte entre la clase media de la sociedad ecuatoriana (niveles socioeconómicos B y C en base a la encuesta de estratificación social realizadas por el Instituto Nacional de Estadísticas y Censos en el año 2011).



## ABSTRACT

*This investigation project explains in detail the development of an electronic system capable of blocking and tracking a motorcycle. Electric starter controls has been digitalized by adding biometric user verification. A tracking system has been included as well. Vehicle's speed is acquired and monitored too. Arduino has been chosen as the microcontroller-based development platform for the embedded system. Prototype possesses two independent blocks: security, and assisted driving. Security block uses an EasyVR 2.0, a reliable user dependent voice recognition module. It also includes a GSM/GS module, based on the SIM908 system-on-chip. Position coordinates can be retrieved and remote engine shutdown has been set. Assisted driving block acquires the motorcycle's speed in order to suggest the better shift for the engine. This way, gas consumption can be bounded. Vehicle's security has been greatly improved as well as gas consumption has been reduced. The development of this embedded system has greatly improved factory electronics of this vehicle. Motorcycles are nowadays popular, reliable and economic rides.*

## Glosario de términos

- ▶ **3G**  
Referencia breve a la tercera generación de las tecnologías de comunicaciones móviles.
- ▶ **ATMEL**  
Fabricante de semiconductores, ampliamente conocidos por su línea de microcontroladores.
- ▶ **AVR (*Advanced Virtual Risc*)**  
Microcontrolador fabricado por ATMEL con juego de instrucciones reducido (RISC).
- ▶ **CAN (*Controller Area Network*)**  
Protocolo de red empleado en redes industriales.
- ▶ **CDI (*Capacitor Discharge Ignition*)**  
Sistema de encendido eléctrico de motocicletas basado en descargar de condensador.
- ▶ **ECU (*Electronic Control Unit*)**  
Unidad electrónica de control de un vehículo.
- ▶ **EEPROM (*Electrically Erasable Programmable Read Only Memory*)**  
Memoria programable de sólo lectura con escritura eléctrica.
- ▶ **FLASH**  
Sección de memoria de un microcontrolador, dedicada al almacenamiento del código fuente de ejecución.
- ▶ **GPIO (*General Purpose Input Output*)**  
Pin de entrada o salida de propósito general.
- ▶ **GPRS (*General Packet Radio System*)**  
Servicio general de paquetes por radio. Extensión de GSM para transmisión de datos mediante conmutación de paquetes.

- ▶ **GPS (*Global Posicionment System*)**  
Sistema de posicionamiento global.
- ▶ **GSM (*Global System for Mobile Communications*)**  
Sistema global de comunicaciones móviles.
- ▶ **IDE (*Internal Development Environment*)**  
Entorno de desarrollo nativo o plataforma de programación.
- ▶ **IDLE**  
Estado de inactividad o de espera de un circuito electrónico o línea de comunicación.
- ▶ **LCD (*Liquid Crystal Display*)**  
Pantalla de cristal líquido.
- ▶ **OpenSource**  
Denominación común para proyectos llevados a cabo y sostenidos por una comunidad entusiasta, bajo la ideología del software y/o hardware libre.
- ▶ **PCB (*Printed Circuit Board*)**  
Placa de circuito impreso.
- ▶ **PIC**  
Microcontrolador fabricado por Microchip con juego de instrucciones reducido (RISC).
- ▶ **REV**  
Tecnico. Revisión. Versión actualizada y examinada de una placa electrónica.
- ▶ **Shield**  
Anglicismo. Denominación común los módulos desarrollados a manera de SoC para placa electrónicas.
- ▶ **SIM (*Suscriber Identity Modulo*)**  
Tarjeta identificadora de abonado GSM.
- ▶ **SoC (*System On Chip*)**

Circuito integrado que agrupa una o más características de un ordenador, con características más potentes que las de un microcontrolador.

- ▶ **UART (*Universal Asynchronous Receiver Transmitter*)**  
Transmisor receptor asíncrono universal. Controlador de puertos y dispositivos serie.
  
- ▶ **TTL (*Transistor – Transistor Logic*)**  
Lógica transistor – transistor. Más conocida como lógica de 5V.
  
- ▶ **USB (*Universal Serial Bus*)**  
Controlador de bus tipo serie universal.

# INTRODUCCIÓN

El presente proyecto estudia las generalidades del sistema eléctrico de una motocicleta, así también como los diagramas de su sistema de encendido y el circuito del velocímetro. El análisis de los diagramas previamente mencionados, permite diseñar un sistema dividido en dos bloques principales: uno de seguridad, y otro de conducción asistida. El bloque de seguridad digitaliza los controles mecánicos de ignición del motor mediante la inclusión de una etapa de verificación biométrica de usuario. También incluye un dispositivo de rastreo mediante GPS. El bloque de conducción asistida realiza la lectura de la velocidad de circulación del vehículo, para sugerir una marcha que permita ahorrar el combustible consumido por la motocicleta.

Se analiza detalladamente información técnica para la selección de equipos afines basados en el uso de sistemas microcontroladores para la implementación de un prototipo de sistema de control de seguridad física y conducción asistida de una motocicleta. Se buscan los elementos necesarios para un sistema de reconocimiento de voz, rastreo mediante GPS y un sistema de conducción asistida; para de esta manera dotar de un valor agregado al vehículo.

Se procede a la modelación de las etapas para el desarrollo del prototipo. Se realiza el estudio de controles, circuitos y sensores a ser intervenidos. El desarrollo del prototipo va acorde a los parámetros determinados por experimentación y referencias bibliográficas.

El presente proyecto de investigación se divide en 5 capítulos que son:

## **1. Capítulo 1 - El problema**

Este capítulo permite conocer desde un contexto macro hasta un contexto micro la situación actual de la industria de la motocicleta a nivel nacional e identifica la problemática debido a la cual se dio la necesidad de desarrollar el presente proyecto de investigación.

## **2. Capítulo 2 – Marco teórico**

En este capítulo se recopila información variada que permite comprender la electrónica embebida en una motocicleta, así también como los grados de integración de los sensores y la evolución de los sistemas electrónicos a lo largo de las últimas décadas en la industria automotriz.

Paralelamente se detalla la información sobre plataformas de desarrollo basadas en sistemas microcontrolados.

## **3. Capítulo 3 – Metodología**

Este capítulo muestra los diferentes métodos que fueron utilizados para la realización del proyecto. Se ha consultado en fuentes bibliográficas como libros, manuales, guías de reparación, información disponible en línea, así también como la consulta con personal especializado en motocicletas.

## **4. Capítulo 4 – Desarrollo de la propuesta**

Este capítulo da a conocer de manera detallada y elaborada todas las actividades relacionadas así como parámetros involucrados para el desarrollo de un sistema de control de seguridad física y conducción asistida para motocicletas. Análogamente se exponen criterios de selección de equipos y las configuraciones necesarias a tomar en cuenta para la puesta en marcha del proyecto.

## **5. Capítulo 5 - Conclusiones y Recomendaciones**

Este capítulo suministra información basada en conclusiones y recomendaciones a considerar en el desarrollo de futuras investigaciones relacionadas con el tema.

# **CAPÍTULO I**

## **EL PROBLEMA**

### **1.1. TEMA**

“Sistema electrónico para Control de Seguridad Física y Conducción Asistida de una Motocicleta”.

### **1.2. PLANTEAMIENTO DEL PROBLEMA**

#### **1.2.1. CONTEXTUALIZACIÓN**

Wilhelm Maybach y Gottlieb Daimler construyeron una moto con cuadro y cuatro ruedas de madera y motor de combustión interna en 1885, su velocidad era de 18 km/h y el motor desarrollaba 0,5 caballos de fuerza. Gottlieb Daimler usó un nuevo motor inventado por el ingeniero Nikolaus Augusto Otto en 1876. El motor de Otto era más liviano y brindaba una potencia ligeramente superior. Estas mejoras permiten que para el año de 1897 se comiencen a fabricar motocicletas con el motor sobre la rueda delantera.

La aparición del Ford – T, el primer vehículo producido en masa y la producción de los primeros utilitarios, obligó a los fabricantes de motocicletas innovar en su diseño. Así es que se incorpora el Sidecar, mejor conocido como triciclo o moto de 3 ruedas. Para el final de la primera guerra mundial, la motocicleta ya gozaba de amplia aceptación y gran popularidad. Estos factores atraen el interés de más empresas automotores que inician su fabricación; bajo sus propios modelos. Este es el inicio de la era de la motocicleta tal y como se la conoce hasta hoy en día [1].

Actualmente en Latinoamérica, grandes y reconocidas marcas fabricantes de motocicletas como BMW, Suzuki, Yamaha, Kawasaki; poseen sus concesionarias para la distribución de sus automotores. Si bien este tipo de vehículo ha evolucionado notablemente con el paso de los años en cuanto a mecánica se refiere, utilizando motores más potentes, suspensiones más estables y frenos ABS (*Anti Block System*, Sistema Anti Bloqueo); debido a su concepto y dimensiones, este vehículo es considerando un blanco vulnerable frente a posibles atracos. Toda implementación en cuanto a seguridad y sistemas de conducción asistidos se refiere, han sido foco principal de la industria del automóvil. BMW con su marca Motorrad ha sido siempre pionera en la inclusión de la electrónica para sus motocicletas, trabajando con la incorporación de instrumentos digitales y sistemas GPS. Pese a su versatilidad, la protección física del vehículo y el desarrollo de sistemas asistidos de conducción, siguen siendo aspectos débiles de las motocicletas.

En Ecuador, existen conocidos distribuidores de motocicletas de origen chino ensambladas nacionalmente, siendo Motor Uno y Shineray claros ejemplos de marcas populares y ampliamente distribuidas. Modelos de estas marcas ofrecidos por las diversas casas comerciales a nivel local no poseen medios de protección física o sistemas de conducción asistidos. La carencia de un medio de protección física puede derivar directamente en pérdidas por robo, debido a que la motocicleta por su peso manejable, es un blanco fácil para la delincuencia. Esto puede conllevar a la pérdida total de la inversión realizada en la compra del vehículo. La carencia de un sistema de conducción asistida puede acarrear consecuencias tales como: decremento de la vida útil del motor y pérdidas económicas en reparaciones, problemas de índice personal en atrasos debidos a fallas técnicas e incluso la paralización de actividades laborales; en caso de que la motocicleta sea parte activa de una empresa.

Consecuencias que sin duda, todo propietario de este vehículo estaría dispuesto a evitar por cualquier medio posible.

### **1.3. DELIMITACIÓN DE CONTENIDOS**

**Área académica: Física y Electrónica**

**Línea de investigación: Sistemas Electrónicos**



## **Sublínea de investigación: Sistemas Embebidos**

### **DELIMITACIÓN ESPACIAL**

La presente investigación se desarrolló en la ciudad de Ambato.

### **DELIMITACIÓN TEMPORAL**

Una vez aprobada por el Honorable Consejo Directivo de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, la presente investigación se llevó a cabo desde el 26 de Junio de 2014 hasta el 29 de Mayo de 2015.

#### **1.4. JUSTIFICACIÓN**

El principal propósito de desarrollo de la presente investigación está enfocado a la carencia de un sistema electrónico de bloqueo de encendido y un sistema de conducción asistida en las motocicletas comercializadas a nivel nacional por la marca Motor Uno.

La inclusión de componentes electrónicos en sistemas puramente mecánicos, facilitan en gran medida la operación, administración y mantenimiento de los mismos. Y es que no resulta fácil para la mayoría de gente conducir una motocicleta. Ciertamente es un proceso que requiere de más instinto y experiencia. Un sistema de conducción asistido puede facilitar la tarea más tediosa para conductores novatos, el cambio adecuado de marchas.

Únicamente ciertos modelos de automóviles de gama alta incluyen sistemas de inicio por contraseñas o algún sistema de ignición distinto al tradicional por llave. Motivo por el cual la inclusión de este tipo de sistema en un vehículo mucho más susceptible a robos, resulta ser una idea práctica e innovadora.

No es una novedad el hecho de que los vehículos sean uno de los blancos preferidos de la delincuencia. El peso manejable y dimensiones reducidas de la motocicleta, hacen que este tipo de vehículo sea un blanco aún más susceptible al hurto. Resulta óptimo incluir opciones de recuperación de posición en caso de que incidentes desafortunados como robos incidan en un vehículo que pueda incluso llegar a ser indispensable para sustento de personas o familias con un nivel socioeconómico B o C.

La innovación en un sector de la industria automotriz que acoge el interés de un amplio mercado en crecimiento y los argumentos mencionados y detallados con anterioridad, aseveran que el desarrollo del presente proyecto resulte factible. Los beneficiarios directos son los potenciales compradores y actuales propietarios de motocicletas de la marca Motor Uno, ya que accederán a un medio de respaldo y protección de su inversión, que garantice la integridad física y buen funcionamiento de su vehículo y/o herramienta de trabajo.

## **1.5. OBJETIVOS**

### **1.5.1. OBJETIVO GENERAL**

- ▶ Implementar un sistema electrónico para el control de seguridad física y conducción asistida de una motocicleta, adaptable a los modelos comercializados a nivel nacional por la marca Motor Uno.

### **1.5.2. OBJETIVOS ESPECÍFICOS**

- ▶ Analizar el sistema electrónico actual incorporado en las motocicletas distribuidas a nivel nacional por la marca Motor Uno.
- ▶ Determinar los requerimientos técnicos para el desarrollo de sistema electrónico de control de seguridad física y conducción asistida de una motocicleta.
- ▶ Diseñar un prototipo de sistema electrónico de control de seguridad física y conducción asistida de una motocicleta para brindar seguridad en la protección de este vehículo.

## **CAPÍTULO 2**

### **MARCO TEÓRICO**

#### **2.1. ANTECEDENTES INVESTIGATIVOS**

Realizada la respectiva investigación bibliográfica en los repositorios digitales de las principales universidades y escuelas politécnicas a nivel nacional, se han encontrado los siguientes trabajos de investigación relacionados:

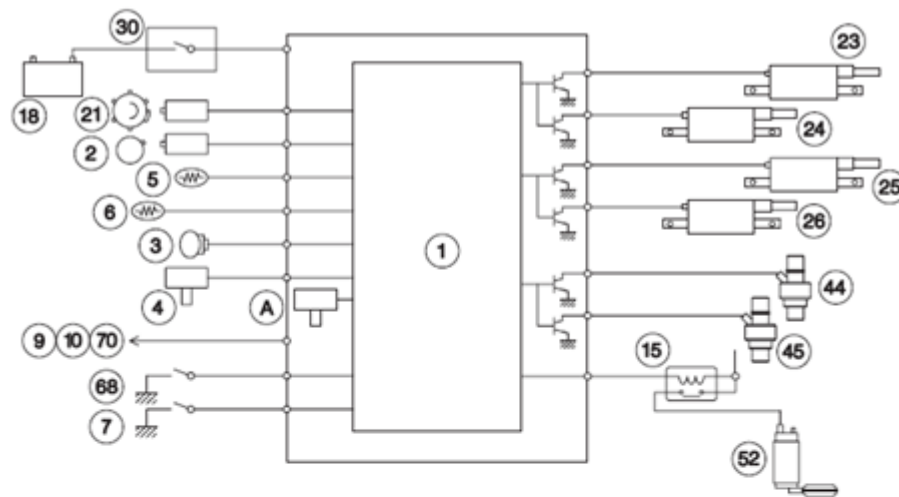
- ▶ Luis Cando, desarrolló un sistema de bloqueo de encendido de vehículos. En su proyecto de investigación se trabaja principalmente con un sensor biométrico de huella dactilar y un microcontrolador PIC16F877A, para incrementar las seguridades frente a posibles escenas de robo. [2]
- ▶ Pantoja Tello y Alexandra Doris, desarrollaron un sistema de conducción asistida para ser incorporado a un automóvil manejado por personas con discapacidad de desplazamiento. Se utiliza principalmente sensores y microcontroladores. [3]

#### **2.2. FUNDAMENTACIÓN TEÓRICA**

##### **2.2.1. SISTEMAS ELECTRÓNICOS DE LAS MOTOCICLETAS**

La ECU (Electronic Control Unit, Unidad Electrónica de Control), es el sistema electrónico gobernante de todo vehículo moderno. La ECU se encuentra conformada por un conjunto de sensores que registran la actividad y condiciones de operación del

motor, que envían señales digitales a uno o a un bloque de sistemas micro procesados [4]. Físicamente su característica más destacable es su cuerpo de aluminio con su respectivo sistema de refrigeración mediante disipadores. En la figura 2.1 se aprecia el diagrama de una ECU perteneciente a un modelo deportivo y sus respectivas conexiones a todos los sensores y actuadores del sistema.



Legend:

- |                                 |                                   |
|---------------------------------|-----------------------------------|
| A) Intake pressure sensor       | 21) Driving shaft position sensor |
| 1) Electronic unit              | 23) Rear cylinder coil "1"        |
| 2) Camshaft position sensor     | 24) Rear cylinder coil "2"        |
| 3) Throttle valve potentiometer | 25) Front cylinder coil "1"       |
| 4) Intake pressure sensor       | 26) Front cylinder coil "2"       |
| 5) Coolant thermistor           | 30) Ignition switch               |
| 6) Air thermistor               | 44) Front cylinder injector       |
| 7) Fall sensor                  | 45) Rear cylinder injector        |
| 9) Clutch control lever switch  | 52) Fuel pump                     |
| 10) Neutral gear switch         | 68) TEST connectors               |
| 15) Fuel pump relay             | 70) Side stand switch             |
| 18) Battery                     |                                   |

Figura 2.1: Diagrama de conexiones del ECU de una motocicleta Aprilia

**Fuente:** Wiring Diagrams [5].

El funcionamiento de la ECU cumple con la lógica de cualquier sistema electrónico, los actuadores son activados en base a las lecturas obtenidas por parte de los sensores, en base a parámetros interpretados por el microprocesador [6]. Debido a la gran cantidad

de elementos que requieren ser conectados a la misma, la ECU dispone de un conector múltiple paralelo, en el cual se indica el respectivo orden de las señales de entrada.

En la siguiente lista se enumeran algunas de las funciones principales de la unidad electrónica de control.

- ▶ Regulación y control del caudal de inyección
- ▶ Medida de gases contaminantes emitidos
- ▶ Recirculación de gases de escape
- ▶ Limitación de presión de sobrealimentación
- ▶ Estrategias de marcha del motor
- ▶ Herramientas de auto diagnóstico y localización de averías
- ▶ Anti arranque codificado

Por ejemplo, una de las estrategias de marcha del motor consiste en la dosificación adecuada del combustible en el instante de la aceleración, el corte de la inyección a un determinado número de revoluciones, el aumento o disminución de combustible en la mezcla mediante el uso de sensores de oxígeno en los escapes, o la asistencia de marchas prescindente de embrague [7].

La siguiente lista detalla los componentes principales de un ECU.

- ▶ Microprocesador maestro o activo: que actúa como elemento principal de procesamiento de señales y toma de decisiones.
- ▶ Microprocesador esclavo: paralelamente se incluye un segundo microprocesador con la funcionalidad de que actúe como central de respaldo de la unidad maestra.
- ▶ RAM: memoria necesaria para la ejecución ya sea por bloques o por líneas del código activo en los microprocesadores.
- ▶ ROM: memoria que almacena el código de control. también es capaz en algunos casos de mantener una copia temporal de los últimos valores registrados en el sistema.
- ▶ Una memoria no volátil adicional, encargada del registro de los códigos de falla suscitados
- ▶ GPIO's (General Purpose Input Output), pines de entrada y de salida

### **2.2.1.1. EVOLUCIÓN DE LOS SISTEMAS ELECTRÓNICOS EN LA INDUSTRIA AUTOMOTRIZ**

A continuación se realiza una condensación breve de las innovaciones tecnológicas que han formado parte de la evolución de la industria automotriz, a partir de la inclusión de las primeras unidades de encendido electrónico [8].


- ▶ Primera generación (1980-1988): la primera generación surge junto con los primeros sistemas de encendido electrónico (el sistema predecesor era mecánico y se basaba en el uso de platinos). La instrumentación era de tipo analógica y los componentes de características robustas tanto eléctrica como físicamente. Con un esquema de funcionamiento sencillo, el proceso se resumía a unos cuantos sensores y actuadores. Estos sistemas al no poseer microprocesadores no podían ejecutar pruebas de funcionamiento, motivo por el cual las fallas en los elementos serán interpretadas como valores leídos correctamente.
- ▶ Segunda generación (1988-1996): la segunda generación incorpora el uso de un microprocesador. Las opciones de control se amplían notablemente y se aumenta el número de sensores y actuadores. Se incluye la herramienta de autodiagnóstico, el cual funciona mediante una serie preestablecida de códigos interpretados en una sección del tablero de control.
- ▶ Tercera generación (1996-2007): la tercera generación presenta sistemas que incorporan más de un microprocesador y existe la posibilidad de reprogramación de los mismos. Esta función se implementa con la visión de los fabricantes de encontrar nuevas configuraciones óptimas de funcionamiento. La introducción de la opción de reprogramación evita que sea necesario reemplazar la unidad entera ante un caso de falla, tal y como solía hacerse en la generación predecesora.
- ▶ Cuarta generación (desde el 2000, redes CAN): la cuarta generación utiliza una red común industrial de tipo CAN, la cual es encargada de enlazar a todos los dispositivos electrónicos. Se basa en la evolución de los sistemas basados en microprocesadores, pero con la utilización de protocolos optimizados, mayor velocidad de procesamiento, medidas de seguridad y tolerancia y corrección de errores.

- ▶ Quinta generación (híbridos): la quinta generación posee un elevado número de instrumentación, herramientas de monitoreo, bloqueo y sistemas de conducción asistida. Mediante el empleo de estos sistemas, se pretende convertir al vehículo en un sistema autosuficiente, en el cual se pueda prescindir del conductor.

### 2.2.1.2. SISTEMA ELECTRÓNICO DE LAS MOTOCICLETAS MOTOR 1

Se procede a analizar uno de los modelos que se encuentra actualmente en producción por parte de Motor Uno. El nombre clave del modelo es Dakar. Las características tanto generales como eléctricas del modelo se pueden apreciar en la tabla 2.1.

Tabla 2.1: Resumen de características Dakar 200cc

<b>IMAGEN</b>		
<b>CARACTERÍSTICAS</b>	<b>GENERALES</b>	<p>Cilindraje: 193cc          Motor de 4 tiempos y 2 válvulas          Máxima potencia de 11 KW (a 7500 rpm)          Máximo torque de 14 Nm (a 6000 rpm)          Peso en seco 130 KG          Máxima capacidad de carga: 150 KG</p>
	<b>ELÉCTRICAS</b>	<p>Sistema de arranque CDI          Batería de 12V – 6.5 AH          Alternador trifásico de imanes permanentes          Fusible de 12A</p>
	<b>INSTRUMENTOS</b>	<p>Tacómetro Análogo          Velocímetro Digital          Odómetro Digital          Sensor de marchas de posición          Control de encendido en base a marcha embragada</p>
<b>Fuente:</b> Dakar User's Manual [9].		

Realizada la inspección respectiva de la circuitería interna del modelo de la tabla 2.1, se ha llegado a determinar que el tipo de ECU incorporada por las motocicletas ensambladas por Motor Uno corresponde a un modelo de segunda generación. Esto se debe a dos criterios principales:

- ▶ El sistema utiliza un solo microprocesador como unidad maestra.
- ▶ El sistema dispone de un reducido número de sensores (velocímetro, tacómetro, indicador de marchas). Estos sensores son de tipo simple, es decir, no cuentan con unidades de autoajuste o autocorrección de datos.

Estas características son expandibles mediante el criterio de inducción general a todos los modelos actualmente ensamblados por Motor Uno a nivel Nacional, ya que se ha verificado que las variantes entre los diferentes modelos radican de manera casi exclusiva en el aspecto físico de la motocicleta.

### **2.2.2.        SENSORES EN LA INDUSTRIA AUTOMOTRIZ**

Los vehículos actuales (especialmente los automóviles) disponen de una gran cantidad de sensores. Los mismos son necesarios para la gestión de los procesos relacionados a algún tipo de instrumentación y todas las señales provenientes de los mismos se dirigen al ECU (Electronic Control Unit, Unidad de Control Electrónico) [10].

En el caso de las motocicletas, que carecen de ECU, las señales de los sensores son procesadas internamente por subsistemas dedicados en el tablero de control.

Los sensores en un vehículo están ubicados usualmente en puntos relativamente cercanos a la generación de las variables a medir o registrar. Este arreglo provoca que los sensores estén expuestos a una serie de factores que pueden alterar o corromper su funcionamiento. Además están en constante sometimiento a cargas extremas, por lo cual deben resistir condiciones igual de apremiantes, así como se puede apreciar a continuación en la tabla 2.2.



Tabla 2.2: Clases de factores externos que debe resistir un sensor automotriz

TIPO DE FACTOR	DESCRIPCIÓN
<b>Mecánico</b>	<ul style="list-style-type: none"> <li>▶ Golpes</li> <li>▶ Vibraciones</li> </ul>
<b>Ambiental</b>	<ul style="list-style-type: none"> <li>▶ Temperatura</li> <li>▶ Humedad</li> </ul>
<b>Químico</b>	<ul style="list-style-type: none"> <li>▶ Salpicaduras de fluidos corrosivos y no corrosivos</li> <li>▶ Niebla salina</li> </ul>
<b>Electromagnético</b>	<ul style="list-style-type: none"> <li>▶ Campos magnéticos</li> <li>▶ Cambios de polaridad</li> <li>▶ Sobre tensiones</li> </ul>
<b>Fuente:</b> Sensores en el automóvil [11].	

Los sensores utilizados para su inclusión en vehículos poseen características especiales que los tornan diferentes a los sensores convencionales. En la figura 2.2, se aprecia los requerimientos de los sensores y las medidas de desarrollo tomadas en cuenta por los fabricantes.

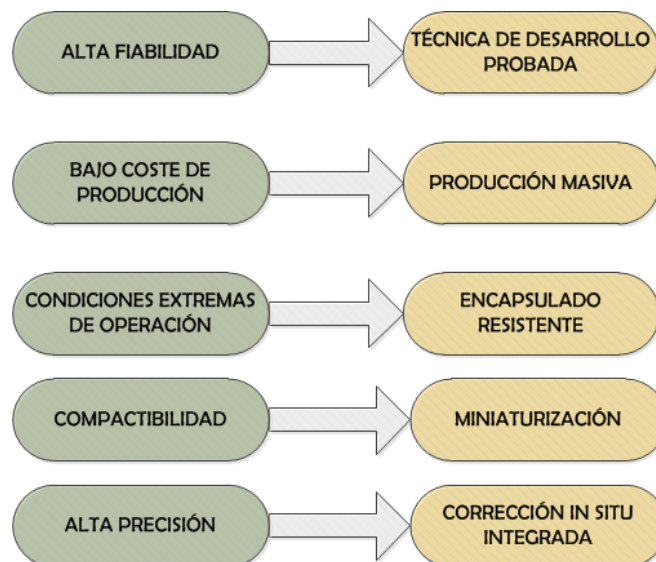


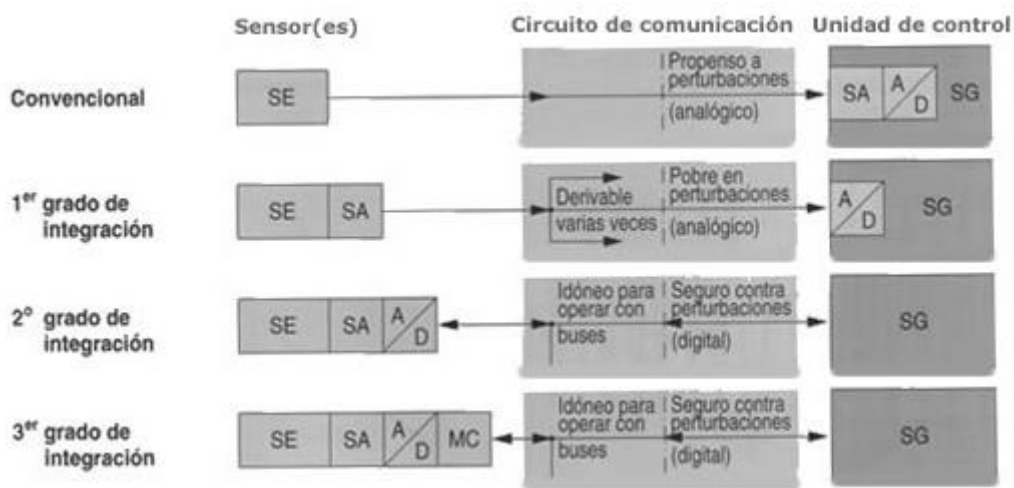
Figura 2.2: Exigencias y medidas de desarrollo de los sensores automotrices

**Fuente:** El investigador, basado en [12].

Se puede tratar a los sensores como elementales o integrados.

- ▶ Sensores elementales: convierten el valor de una variable física en una señal eléctrica que precisa de tratamiento y en algunos casos, también de pre amplificación.
- ▶ Sensores integrados: agrupan en un mismo bloque las funciones del sensor elemental más el procesamiento de la señal, lista para ser procesada mediante algún microcontrolador o sistema embebido.

La figura 2.3 muestra los posibles niveles de integración de un sensor utilizado en la industria automotriz.



SE: sensor elemental  
 SA: acondicionamiento de señal analógica  
 A/D: convertor análogo/digital  
 SG: unidad electrónica de control  
 MC: microprocesador

Figura 2.3: Niveles de integración de un sensor automotriz  
**Fuente:** Los sensores en el automóvil [13].

Resulta común en la práctica la máxima integración dentro de lo posible de los sistemas para evitar conexiones que pueden incurrir en errores de lectura o de funcionamiento. En algunas ocasiones incluso se utilizan sistemas redundantes, como medida de seguridad contra un posible fallo o avería de una de las unidades.

La industria automotriz centra sus esfuerzos en la durabilidad y resistencia de sus sensores, no siento así el caso de la exactitud de los mismos. Comparado con un sensor industrial, uno de tipo automotriz posee una exactitud tolerable (alrededor del 1% de error de medición). Para compensar este error de medición, las señales adquiridas son corregidas mediante técnicas de filtrado y re calibración en la unidad electrónica de control (ECU) [14].

### 2.2.2.1. SENSOR DE EFECTO HALL

El efecto Hall es un fenómeno por el cual los portadores de carga que se desplazan a través de un campo magnético son forzados hacia uno de los extremos del medio de conducción [15].

La figura 2.4 detalla como en presencia de un campo magnético externo, los portadores de carga sufren una distribución inhomogénea sobre la superficie del material conductor.

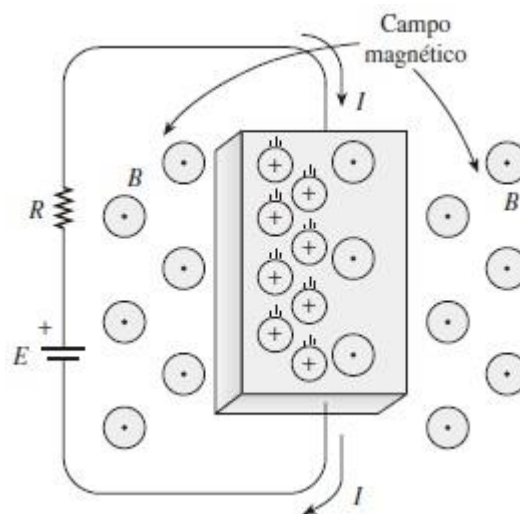


Figura 2.4: Explicación gráfica del Efecto Hall

**Fuente:** Electrónica Industrial Moderna [16].

La magnitud de la fuerza que actúa sobre cada portador de manera individual viene dada por la relación descrita en la ecuación 2.1.

$$F = qvB \quad (2.1)$$

Donde se tiene que:

$v = \text{velocidad en } \frac{m}{sg} \text{ del portador}$

$q = \text{carga en Coulombs}$

$B = \text{densidad de flujo magnético en Teslas o } \frac{wb}{m^2}$

No es necesario que la superficie conductora sea fabricada a partir de un material conductor, en la realidad los sensores de efecto Hall son producidos a partir de finas láminas de material semiconductor. En algunos casos incluso es utilizado cristal de alta resistividad. Para su utilización en instrumentación o cualquier proceso electrónico de control, los sensores de efecto Hall vienen en circuitos integrados, pudiendo ser éstos de dos tipos:

- ▶ Lineal: el valor de la salida varía en forma directamente proporcional a los cambios del flujo magnético aplicado.
- ▶ Umbral: el valor de la señal de salida cae de manera abrupta al presenciarse cierto valor de flujo magnético.

El voltaje de efecto Hall es muy pequeño. Motivo por el cual, cualquier sensor construido debe incluir un bloque amplificador de voltaje de alta ganancia [17].

Un sensor de efecto Hall puede ser utilizado como interruptor de repetición en frecuencias de hasta 100KHz, cuesta menos que los interruptores mecánicos y no posee ninguno de los problemas asociados a los mismos. Este tipo de sensores son comunes en la industria automotriz debido a que pueden operar en condiciones de funcionamiento severas, no presentan desgaste de tipo alguno y son inmunes a la mayoría de efectos ambientales [18]. Constan de 3 elementos principales, cuya disposición física real es ilustrada por la figura 2.5.

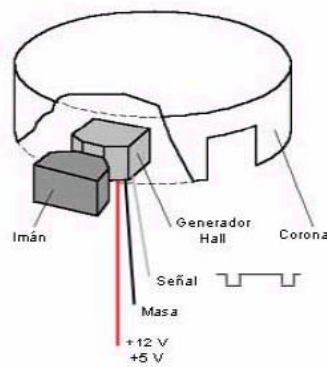


Figura 2.5: Estructura de un sensor de efecto Hall  
**Fuente:** Sensor de Efecto Hall [19].

- ▶ Fuente de generación de flujo magnético: imán permanente.
- ▶ Módulo electrónico: encapsulamiento de los elementos empleados para la detección y medida del voltaje transversal generado.
- ▶ Corona metálica dentada: para la interrupción del campo magnético.

Físicamente el sensor dispone de 3 cables. Dos de ellos componen los terminales de alimentación ( $V_{CC}$  y  $GND$ ), el tercero correspondiente a la señal de salida ( $S_{OUT}$ ).

La corona metálica suele estar intercalada entre el imán fijo y el módulo electrónico de detección de voltaje; unida a un eje con giro. Cuando se produce un movimiento en el eje acoplado, la corona cambia de posición, permitiendo o denegando la detección del campo magnético generado por el imán permanente. El módulo electrónico amplifica la señal a niveles lógicos compatibles:

- ▶ De 0V a 0.5V para un cero lógico (corona blindada).
- ▶ De 5V a 12V para un uno lógico (corona sin blindaje).

Se detallan a continuación los usos del sensor de efecto Hall dentro de la industria automotriz:

- ▶ Sensor de velocidad
- ▶ Detección de posición de: árbol de levas, caja de cambios.
- ▶ Detección de encendido en ciertos modelos de distribuidores.

### 2.2.3. HARDWARE LIBRE

En primera instancia es algo difícil contar con una definición formal de lo que es el Hardware Libre, debido a lo relativamente nuevo en su concepto y aparición; es algo que está en constante evolución [20]. Una de las de las posibles denominaciones más aceptadas por la comunidad consiste en la comparación del mismo con el software libre.

El hardware libre (también llamado “open source” o de fuente abierta), comparte muchos principios y metodologías del software libre. En particular, el hardware libre permite que la gente pueda estudiarlo para entender su funcionamiento, modificarlo, reutilizarlo, mejorarlo y compartir dichos cambios. En la figura 2.6 se puede apreciar una placa de hardware libre.



Figura 2.6: Consola Uzebox  
**Fuente:** The Uzebox Project [21].

Para conseguir dicho fin, la comunidad ha de tener acceso a los ficheros esquemáticos del diseño del hardware en cuestión. Estos archivos, permiten a cualquier persona con los conocimientos y herramientas adecuadas, reconstruir o rediseñar dicho hardware, al conocer el detalle de cada uno de sus elementos componentes y las interconexiones los mismos. En sí, la idea de crear sistemas propios en base a componentes adquiridos por uno o más usuarios, se remonta a la década de los años setenta, donde ya los dispositivos electrónicos (primeros microcontroladores, transistores, etc) ya se encontraban disponibles a la venta. Lo que sí es nuevo es la notable influencia que ha

ido adquiriendo este tipo de hardware frente a las nuevas tendencias educativas y de uso personal, ya sea con fines de lucro o no [22].

Hoy en día, se pueden encontrar en las páginas de cualquier proyecto o comunidad de hardware libre, tres tipos de archivos diferentes:

- **Diseño Esquemático:** muestra la conexión de pines y configuraciones lógicas de cada uno de los componentes de la placa. Sin embargo, no provee información alguna sobre el aspecto físico de la misma.
- **Archivos PCB:** los populares archivos de pistas impresas. Muestra las dimensiones de cada uno de los componentes, así también como el tipo de encapsulado de los mismos. Aunque genera una idea más clara que la del esquemático, aún no se puede determinar las dimensiones físicas reales de la placa.
- **Archivos de Fabricación (Gerber):** este tipo de archivos contienen los datos necesarios para que la placa puede ser replicada mediante maquinaria especializada [23].

La primera tentativa seria de una implementación de hardware libre se da en 1990 con el uso de FPGA's (*Field Programmable Gate Array*, Arreglo de Compuertas Programables). Estos circuitos integrados se basan en compuertas internas programables que pueden desarrollar desde la tarea más sencilla, actuar como un interruptor por ejemplo, a procesos con cierto grado de complejidad. La programación de las mismas se basa en el uso de lenguajes específicos. Dado a que era posible compartir los códigos utilizados para la programación y usando este hecho como punto de partida; Reinold Lamberts y su Open Design Circuits, se consolidan como la primera comunidad web con una visión los principios del software libre aplicados al hardware [24].

Ciertamente lo único que se compartió fueron los códigos de programación. Debido a la ausencia de un software libre para el desarrollo de hardware, no fue posible crear o compartir algún diseño físico de un prototipo de placa real, hablando estrictamente. Sin embargo, la ideología de esta iniciativa, sentó las bases de para lo que estaba por venir.

### 2.2.3.1. ARDUINO

Arduino nació bajo la simple premisa de dotar tanto a aficionados como entendidos de la electrónica, de un sistema microcontrolador completamente flexible y funcional a un precio relativamente bajo. Hoy en día es uno de los proyectos de hardware OpenSource (*Open Source*, de código abierto) más conocido a nivel mundial. El rotundo éxito que ha obtenido este proyecto, hace que uno encuentre disponible una multitud de módulos o shields fabricados por terceros listos para interactuar con las placas de Arduino.

Las placas de Arduino trabajan con microcontroladores AVR (Advanced Virtual RISC, RISC: Reduced Instruction Set Computer) fabricados por ATMEL [25]. La tabla 2.3 muestra las características principales de estos microcontroladores y las placas Arduino que los utilizan.

Tabla 2.3: Microcontroladores utilizados por las placas Arduino

AVR	TIPO	RELOJ	FLASH	EEPROM	SRAM	MODELO
ATMEGA168	ARM 8 bits	16 MHz	16KB	512 bytes	1KB	Duemilanove Diecimila Micro Pro
ATMEGA328	ARM 8 bits	16 MHz	32KB	1KB	2KB	Uno Duemilanove Pro
ATMEGA32u4	ARM 8 bits	16 MHz	32KB	1KB	2.5KB	Leonardo Yun Esplora Lilypad Nano
ATMEGA2560	ARM 8 bits	16MHz	256KB	4KB	8KB	Mega
ATSAM3X8E Cortex-M3	ARM 32 bits	84 MHz	512KB	6KB	100KB	Due
Quark SoC X1000	ARM 32 bits	400 MHz	512KB	11KB	512KB	Intel Galileo
<b>Elaborado por:</b> El Investigador, en base a [26, 27, 28, 29, 30].						

AVR fue un proyecto desarrollado bajo el concepto de la programación en C desde sus inicios. C se basa en la utilización de punteros para los respectivos direccionamientos de memoria. Existen a su vez 5 subdivisiones de la familia AVR de ATMEL, como se puede apreciar en la tabla 2.4.



Tabla 2.4: Clasificación de la familia AVR fabricada por ATMEL

CLASE	FLASH	ENCAPSULADO	JUEGO DE PERIFÉRICOS
ATXMEGA	[16 – 384] KB	44, 64, 100 pines	Extenso con DACS
ATMEGA	[4 – 256] KB	28 a 100 pines	Mediano
ATTNY	[0.5 - 8] KB	6 a 20 pines	Limitado
AT90USB AT90CAN	ATMEGA con controlador USB o controlador CAN respectivamente integrado		
<b>Elaborado por:</b> El Investigador, en base a [31].			

### 2.2.3.1.1. LENGUAJE DE PROGRAMACIÓN ARDUINO

Todas las placas pertenecientes a la familia Arduino se programan mediante el uso de su propio lenguaje de programación. A más de eso, debido a que Arduino utiliza transmisión serial de datos, se pueden recurrir a lenguajes de programación diferentes que admitan este tipo de comunicación.

Una de los rasgos más notorios del lenguaje Arduino es que resulta familiar para todo programador que ha trabajado con lenguaje C o C++. Hereda principalmente los rasgos de C y algunas características de C++. La sintaxis básica, las estructuras de control y el tipo de variables del lenguaje C están soportados completamente. Debido a que Arduino es un entorno de programación basado en hardware, además de las funcionalidades de C, posee sus propias funciones específicas para delimitar entradas y salidas, tanto analógicas como digitales [32]. Se puede decir que Arduino es un híbrido a medio camino entra la sencillez y potencia del lenguaje C y la adaptabilidad de un sistema micro controlado. Un software robusto con una interfaz sencilla, ciertamente resulta de gran atractivo para la realización de cualquier proyecto electrónico

A continuación se detalla una lista de los elementos comunes heredados del lenguaje originalmente desarrollado por Dennis Ritchie.

- ▶ Sintaxis básica: comentarios, cabeceras, delimitadores, operadores aritméticos, operadores booleanos, incrementos y decrementos, entre otros.
- ▶ Estructuras de control: condicionales (if, else), do while, for, switch, break, continue, return.
- ▶ Variables
- ▶ Funciones Matemáticas: min, max, pow, sqrt.

Las funciones matemáticas (así también como el empleo de variables de tipo float) del lenguaje Arduino son considerablemente más potentes que las utilizadas por otros lenguajes de programación para microcontroladores.

### **2.2.3.1.2. AVR VS PIC**

No se puede realizar una comparativa entre Arduino y PIC, ya que Arduino es una plataforma de desarrollo basada en un microcontrolador, mientras que PIC es solamente el último elemento citado. Pero sí se puede realizar una comparación en los microcontroladores empleados por Arduino, fabricados por ATMEL frente a la familia PIC de Microchip [33].

Al igual que con PIC, ATMEL cuenta con una comunidad en línea para el desarrollo de proyectos electrónicos y que se encuentra en constante crecimiento, por varias razones que son citadas a continuación:

- ▶ Índice de alta compatibilidad entre las familias ATXMEGA, ATMEGA y ATTINY.
- ▶ AVR dispone de una arquitectura secuencial para las etapas de carga de código y ejecución de código. Dicha característica le permite ejecutar gran parte de las instrucciones del programa en apenas un ciclo de reloj. Motivo por el cual AVR es relativamente veloz en comparación a otros microcontroladores de 8 bits.
- ▶ AVR dispone de compiladores gratuitos, no siendo este el caso de Microchip.
- ▶ El creciente uso de los microcontroladores ATMEL en placas de hardware libre y la creación de plataformas nuevas de desarrollo, simplifica el proceso de carga y depuración de código.
- ▶ La arquitectura interna de AVR, separa el bus de instrucciones del bus de datos.

- ▶ El stack (*stack*, pila) de PIC es una sección de memoria aislada de la memoria del programa y de la memoria de datos. En un AVR el stack se localiza en la unificación de los 32 registros de entrada – salida con la memoria de datos. AVR diferencia de PIC no posee un stack limitado a un tamaño fijo [34].

#### 2.2.4. MÓDULOS GSM GPS

El surgimiento de plataformas de desarrollo basadas en sistemas microcontrolados, ha generado la aparición de módulos expansibles a con cierta similitud con SOC's (*System On Chip*, Sistema Sobre Circuito) para agregar funciones adicionales a la placa huésped.

Para las opciones de desarrollo adaptables sistemas embebidos, existen los conocidos “shields” (módulos), que poseen una disposición física adaptable y compatible al pinout de la placa de la plataforma de desarrollo original, tal y como se aprecia en la figura 2.7.



Figura 2.7: Módulo GSM/GPS para Arduino  
**Fuente:** GPRS/GSM Quadband Module for Arduino [35].

Los shields de tipo GSM-GPS permiten añadir características de rastreo y conectividad de red.

Dentro de las características principales de estos módulos, se tiene [36]:

- ▶ Circuito integrado capaz de soportar comunicaciones de tipo cuatribanda (850-900-1800-1900 MHz).
- ▶ Compatibilidad GPRS (General Packet Radio System) .
- ▶ Unidad de tipo receptora para GPS.

- ▶ Antenas de GSM y GPS de alta ganancia.
- ▶ Slot para una tarjeta SIM (Subscriber Identity Module).

Aunque su costo por lo general está encima de las plataformas a las cuales están orientados, últimamente la diversificación y la competencia han reducido sus costes de forma considerable.

Para el funcionamiento de la unidad GPS se extraen los datos de ubicación, tiempo de solicitud y velocidad de desplazamiento del sistema electrónico en el cual se encuentra embebido, mediante estándares, de los cuales NMEA 0183 es el más ampliamente utilizado; ya que garantiza compatibilidad del formato de los datos entre la mayoría de dispositivos receptores.

El consumo de corriente de estos módulos es bastante bajo en modo de reposo (15 -20 mA), pero puede subir de forma considerable (hasta cerca de 1A) al momento de iniciar un posicionamiento en frío o comunicarse con la red GSM [37].

#### **2.2.4.1. PROGRAMACIÓN DE LOS MÓDULOS GSM GPS**

La programación de estos módulos es específica y varía de fabricante a fabricante, aunque existe una base de programación común para este tipo de módulos considerandos routers de propósito general. Este esquema de programación se conoce como comandos AT (AT como Attention – Atención) desarrollados por HAYES Communications en la década de los 90's.

La característica principal de este lenguaje, es la de que todos sus comandos empiezan por las letras AT, motivo por lo cual, se lo conoce también como lenguaje de comandos AT. La sintaxis de un comando incluye como se dijo obligatoriamente en primer lugar el prefijo AT, seguido del comando o comandos subsecuentes, como se puede apreciar en la tabla 2.5.

Tabla 2.5: Listado de comandos HAYES principales

AT	Atención. Debe preceder a todas las órdenes restantes
A	Fuerza el modo respuesta (Answer).
A/	Vuelve a ejecutar la última orden enviada.
B	Selecciona el tipo de modulación del módem.
C	Transmisor activado o desactivado.
D	Marca un número telefónico.
E	Habilita o inhabilita el eco de caracteres a la pantalla.
F	Conmuta entre dúplex y semidúplex.
H	Cuelga o descuelga el teléfono.
I	Petición del código de identificación o petición de la suma de control.
L	Selección del volumen del altavoz.
M	Activa o desactiva el altavoz.
N	Negociación de la velocidad.
O	Volver a conexión en línea.
P	Marcación de un número mediante el sistema de pulsos.
Q	Petición del módem para enviar o inhibir el envío de códigos de resultados.
R	Cambia las frecuencias del módem de originador a destinatario de la llamada.
S	Fija el valor del registro.
T	Marcación de un número mediante el sistema de tonos (multifrecuencia).
V	Devolver los códigos de resultado con palabras o números.
W	Selección de los mensajes de progreso de la negociación
X	Uso del conjunto de códigos de resultado básico o extendido.
Y	Habilitar o inhabilitar la desconexión por espacios largos.
Z	Reiniciar módem.
+++	Comando escape.
<b>Fuente:</b> Juego de Comandos Hayes [38].	

La normativa ITU-T V.250 (ITU: International Telecommunications Union, Unión Internacional de comunicaciones, la última revisión del documento citado fue realizada

en 07/2003), sobre Marcado Serial Asíncrono Automático, define las parametrizaciones de los comandos AT básicos.

### **2.2.5. SISTEMAS DE IDENTIFICACIÓN BIOMÉTRICOS**

En primera instancia, se hace necesario aclarar que la Biometría es la ciencia que realiza la cuantización estadística de las características biológicas de un individuo.

La automatización de la Biometría deriva en la Biometría Informática. La aplicación de ésta resulta en los sistemas Biométricos. Existen principalmente 4 requisitos indispensables para que una característica pueda ser considerada como biométrica [39].

- ▶ **Universalidad:** cualquier individuo debe poseerla.
- ▶ **Permanencia:** la característica permanece prácticamente inalterable al tiempo o circunstancia.
- ▶ **Unicidad:** la característica es única en cada individuo, y ésta no puede ser replicada.
- ▶ **Cuantificación:** la característica puede ser medida.

El prescindir de una contraseña o código que puede ser vulnerado mediante múltiples métodos, o en la mayoría de los casos olvidado, hace que el uso de una característica biométrica como identificación al acceso de cualquier sistema sea una alternativa interesante.

#### **2.2.5.1. PROCESO DE RESOLUCIÓN DE IDENTIDAD**

La biometría se basa en 2 procesos para resolver la identidad de un individuo a través de una de sus características biométricas, el reconocimiento y la verificación. El reconocimiento se basa en cuantizar la característica en base a algoritmos para ser posteriormente almacenada en bloques de memoria o bases de datos. La verificación es el proceso comprendido entre adquirir una nueva muestra y compararla con el registro ya existente. En la figura 2.8 se puede apreciar el proceso previamente descrito.

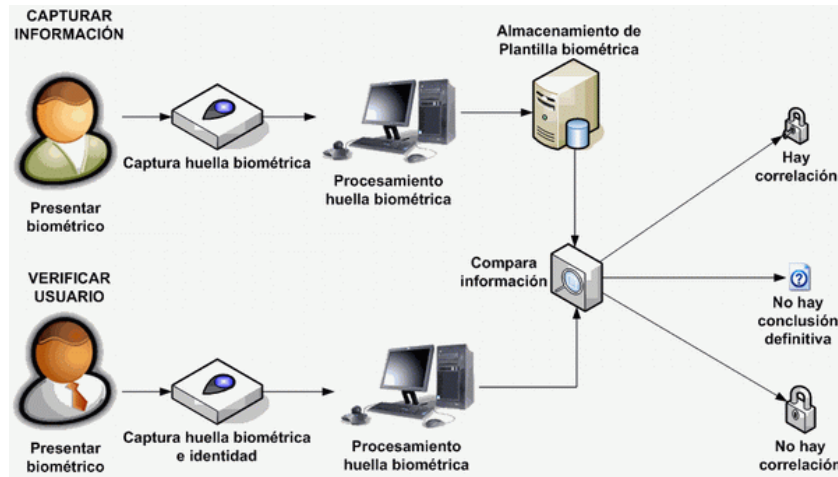


Figura 2.8: Procesos biométricos de reconocimiento y verificación  
**Fuente:** FRAX Biometría [40].

La toma de decisiones en un sistema de identificación biométrico puede derivar en tres condiciones:

- ▶ **Correlación:** el dato biométrico capturado con la(s) plantilla(s) almacenada(s) en el registro o en la base se encuentra dentro de los umbrales de coincidencia. Se concede el acceso.
- ▶ **Correlación nula:** el dato biométrico capturado con la(s) plantilla(s) almacenada(s) en el registro o en la base se encuentra fuera de los umbrales de coincidencia. Se deniega el acceso.
- ▶ **Discrepancia:** la información proporcionada al sistema para poder hacer una comparación adecuada no es suficiente. Se deniega el acceso.

Una de las características principales de los sistemas biométricos es su alta fiabilidad. Si bien, nunca se descarta la posibilidad de una respuesta inconclusa por parte del sistema; el índice de ocurrencia de las mismas es mínimo y en la mayoría de veces se debe a descuidos o condiciones anormales en el usuario.

#### 2.2.5.2. TIPOS DE SISTEMAS BIOMÉTRICOS

Existe una clasificación simple pero obvia que separa a dos grandes grupos para el uso de tecnologías biométricas. Éstas pueden ser estáticas o dinámicas. Para simplificar el asunto, no se dirá más que un sistema estático requiere una verificación de usuario en la

cual, el mantenimiento de la posición inicial es un factor determinante. Los sistemas dinámicos requieren que el usuario demuestre en movimiento alguna característica propia, como el modo de caminar por ejemplo [41].

### 2.2.5.3. TECNOLOGÍAS BIOMÉTRICAS ESTÁTICAS

La tabla 2.6, resume las principales tecnologías estáticas implementadas en la mayoría de sistemas de identificación biométricos.

Tabla 2.6: Tabla de resumen de Tecnologías Estáticas

<i>Tecnología</i>	<i>Descripción Breve</i>	<i>Seguridad</i>	<i>Implementación</i>
<i>Facial</i>	Escaneo de la geometría del rostro del usuario	Alta	Sencilla
<i>Dactilar</i>	Lectura de los patrones de la huella digital	Baja	Sencilla
<i>Geometría de la mano</i>	Técnica en 3D que analiza patrones como largo de los dedos	Media	Sencilla
<i>Escáner de iris</i>	Sucesor altamente confiable del escáner de retina	Alta	Media
<i>Firma Digital</i>	Técnica variable, basada en la determinación de puntos de inflexión de la firma	Media	Media
<i>Voz</i>	Análisis del espectro de frecuencia de la voz humana	Alta	Sencilla
<i>Termografía Facial</i>	Tecnología térmica que busca puntos de calor	Media	Media
<i>ADN</i>	Seguramente los más confiables y precisos. Registro de la cadena de ácido desoxirribonucleico del usuario.	Alta	Compleja
<b>Fuente:</b> Sistemas Biométricos [42].			



### 2.2.5.3.1. RECONOCIMIENTO DE VOZ

Este tipo de sistema de identificación biométrica consiste en la conversión de una señal analógica de baja frecuencia, como es la voz, a una señal digital comprensible y almacenable para un sistema micro procesado, micro controlado o un ordenador.

El objetivo del reconocimiento de voz es la conversión fiel de palabras a texto. Sin embargo, todo sistema reconocedor posee una tasa de error asociada, basada en la suma de tres conceptos [43]:

- ▶ Sustituciones: o palabras que han sido reconocidas erróneamente debido a la similitud entre ambas.
- ▶ Inserciones: reconocer “*la computadora*” en lugar de “*computadora*”. Aumento de palabras en el proceso de reconocimiento.
- ▶ Elisiones: reconocer “*computadora*” en lugar de “*la computadora*”. Eliminación de palabras en el proceso de reconocimiento.

La tabla 2.7 resume las ventajas y desventajas de esta tecnología biométrica.

Tabla 2.7: Ventajas y Desventajas del reconocimiento de voz

<i>Ventajas</i>	Facilidad de medida: el hardware requerido es económico y la implementación es sencilla.
	Alta universalidad: el déficit de habla en la población mundial es reducido.
	Buenas prestaciones: la verificación requiere de bajos recursos de procesamiento y el volumen de información almacenada es más que aceptable.
	Alta aceptabilidad: debido a que el sistema no es intrusivo, los usuarios no se cohíben de utilizarlo.
<i>Desventajas</i>	Permanencia Media: los parámetros básicos de la voz pueden verse afectados por diversos factores.
	Unicidad Media: no resulta extraño encontrar casos de parámetros vocales altamente similares.
	Seguridad media: una grabación puede garantizar acceso a usuarios no autorizados.
<b>Elaborado por:</b> El Investigador, basado en [44].	

Se detallan a continuación las principales características del reconocimiento de voz.

- ▶ De alta sensibilidad con respecto a factores externos (ruido ambiental), estado de ánimo, envejecimiento y enfermedades en el usuario.
- ▶ Sistema de tipo no intrusivo.
- ▶ Configuración simple, la matriculación toma pocos minutos.

El reconocimiento de voz posee una excelente tasa precisión – número de componentes. Por lo general solo basta con un micrófono, una etapa de pre amplificación y un sistema micro controlado. Además cuenta un alto índice de universalidad, debido a que puede ser utilizado para trabajar con cualquier lenguaje y resulta adaptable a la mayoría de sistemas.

#### 2.2.5.3.1.1. PROCESO DE RECONOCIMIENTO DE VOZ

Los sonidos son digitalizados mediante el uso de micrófonos y generalmente sistemas piezoeléctricos. El material piezoeléctrico responde ante frecuencias audibles, generando pequeñas tensiones que pueden ser amplificadas. Estas tensiones son muestreadas y convertidas a señales digitales compatibles con la unidad central de procesamiento del sistema de reconocimiento de voz.

Un diagrama de bloques simplificado del proceso de reconocimiento de voz, puede ser apreciado en la figura 2.9.

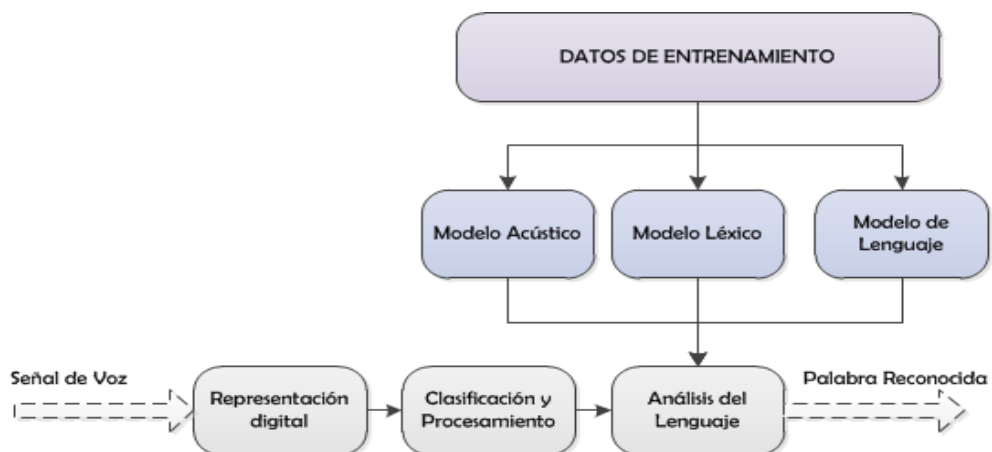


Figura 2.9: Diagrama de Bloques de un sistema de reconocimiento de voz

**Fuente:** El investigador, basado en [45].

## 2.2.6. INTERFACES DE SALIDA

Una interfaz se define como un punto de convergencia entre dos o más elementos de distinta naturaleza. La expansión de este concepto a cualquier tipo de sistema electrónico, determina que una interfaz es el medio de comunicación entre el usuario y el sistema propiamente dicho. Una interfaz de salida, muestra los resultados obtenidos por parte del procesamiento interno hacia el usuario de una manera sensorialmente entendible y asimilable.

Las interfaces de salida informan al usuario si es pertinente sobre el estado de procesamiento de los datos a nivel interno del sistema, o sirven para reflejar la toma de decisiones mediante el control de variables o cambio de una entrada.

### 2.2.6.1. TIPOS DE INTERFACES DE SALIDA

#### *INTERFACES GRÁFICAS*

Las interfaces de salida gráficas son las más utilizadas, se encuentran presentes en casi la totalidad de cualquier tipo de sistema electrónico, debido a su versatilidad y facilidad de interpretación. La utilización de displays viene a ser la opción más popular y económica, para sistemas con bajos requerimientos de procesamiento y de entrega de datos. En la figura 2.10 se aprecia un display popular para el desarrollo de proyectos electrónicos basados en sistemas microcontrolados.

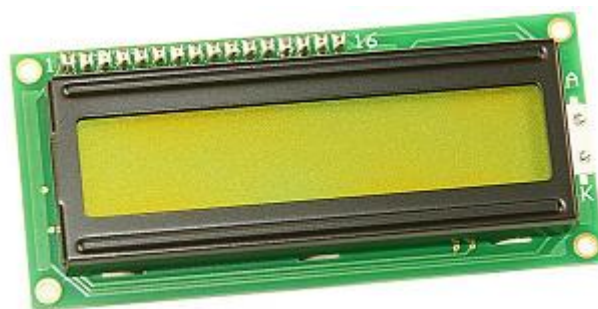


Figura 2.10: Display LCD de 2x16  
**Fuente:** Microcontroladores PIC [46].

Este tipo de pantallas de cristal líquido es relativamente sencillo de controlar, sin embargo es necesario tomar en cuenta algunas consideraciones para el manejo correcto de este tipo de interfaces para la correcta interacción con cualquier sistema microcontrolado:

- ▶ Librerías o sintaxis: que varía de fabricante a fabricante y las mismas deben ser incluidas para que el controlador interno (en caso de poseerlo), interprete las órdenes que han sido enviadas.
- ▶ Consumo de corriente: la mayoría de displays funcionan con corrientes sumamente bajas, pudiendo ser conectados directamente a los pines del microcontrolador o microprocesador en cuestión. Sin embargo en algunos casos es necesaria una etapa transistorizada para abarcar las necesidades de alimentación.
- ▶ Tiempo de arranque: parámetros a revisar en las hojas de datos provistas por los fabricantes de displays con controladores internos [47]. Ciertos modelos pueden arrancar directamente, mientras que otros requieren de pequeños tiempos de inicialización para que el controlador interno cargue complementos asociados al manejo de la pantalla.

### *INTERFACES ACÚSTICAS*

Las interfaces acústicas hacen uso del espectro de audible de frecuencias (0 -20KHz). Ejemplos de este tipo de interfaces son: reproductores MP3, sintetizadores, parlantes, zumbadores piezoeléctricos, etc. Su utilización resulta conveniente en procesos relacionados al reconocimiento de voz.

### *INTERFACES DE TRANSMISIÓN DE DATOS*

Las interfaces de transmisión de datos añaden conectividad no soportada por el sistema nativo para comunicarse con sistemas externos, mediante la utilización de protocolos comunes de comunicación. Se cita como ejemplos del presente tipo de interfaz a módulos de comunicación serial, bluetooth, I2C, SPI, Zigbee, Xbee, entre otros.

## *INTERFACES DE ACTUACIÓN*

Las interfaces de actuación son conocidas también como etapas de amplificación o de potencia. Permiten que el sistema nativo controle o interactúe con elementos cuyas tensiones o corrientes sean muy superiores a los valores de alimentación nominales del sistema. Los actuadores de tipo electromecánicos son los más difundidos debido a su robustez y versatilidad. Se citan como ejemplos: relés, relés de estado sólido, relés térmicos, triacs, opto acopladores, entre otros.

### **2.3. PROPUESTA DE SOLUCIÓN**

Con el desarrollo del sistema electrónico de control de seguridad física y conducción asistida de una motocicleta, se brindará innovación y respaldo a la inversión de los propietarios de este popular medio de transporte.

## **CAPÍTULO 3**

### **METODOLOGÍA**

#### **3.1. MODALIDAD BÁSICA DE LA INVESTIGACIÓN**

La presente fue una investigación aplicada, ya que la misma se desarrolló utilizando los conocimientos adquiridos a lo largo de la formación en la carrera de Ingeniería en Electrónica y Comunicaciones de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato; para comprender, analizar y resolver el problema planteado.

Se ha utilizado la investigación bibliográfica, debido a que la explicación científica de las variables del tema de investigación, la resolución del problema planteado y la propuesta de solución fueron sustentados consultando en libros, revistas, artículos técnicos, artículos tecnológicos y mecánicos, y publicaciones en internet.

Se ha realizado una investigación de campo, ya que la recolección de datos ha sido proveída distribuidores de motocicletas, así también como talleres mecánicos especializados. Las mediciones pertinentes asociadas con las pruebas del prototipo, han sido recogidas mediante pruebas de desempeño en el área designada para dicho fin.

La presente fue una investigación experimental, porque se manipularon variables no comprobadas en ambientes de prueba para la implementación del prototipo.

## **3.2. RECOLECCIÓN DE INFORMACIÓN**

### **3.2.1. PLAN DE RECOLECCIÓN DE INFORMACIÓN**

La recolección de la información se realizó mediante bibliografía seleccionada, revisión de manuales y guías de reparación, información disponible en línea y consultas a personal especializado en motocicletas.

### **3.3. PROCESAMIENTO Y ANÁLISIS DE LA INFORMACIÓN**

La información recolectada se sometió a un análisis crítico para descartar información de poca relevancia para solamente conseguir datos específicos y concretos que ayuden al desarrollo de un prototipo de un sistema electrónico para el control de seguridad física y conducción asistida, adaptable a los modelos de motocicletas comercializados a nivel nacional por la marca Motor Uno.

### **3.4. DESARROLLO DEL PROYECTO**

Para el desarrollo de la investigación se efectuaron los siguientes pasos:

- Recolección de información sobre los sistemas electrónicos actualmente incluidos en motocicletas comercializadas a nivel nacional.
- Procesamiento de información de sistemas electrónicos de baja potencia adaptables al proyecto.
- Recolección de información técnica sobre sistemas de navegación y equipos afines incluidos en vehículos automotores.
- Análisis de la información técnica para seleccionar los equipos a utilizarse en el sistema.
- Diseño de las etapas para desarrollar el prototipo del sistema electrónico empleando la información recolectada.
- Implementación del prototipo de sistema electrónico para el control de seguridad física y conducción asistida de una motocicleta.
- Evaluación y comprobación del funcionamiento del prototipo.

## **CAPÍTULO 4**

### **LA PROPUESTA**

#### **4.1. ANTECEDENTES DE LA PROPUESTA**

Realizada la respectiva investigación tanto bibliográfica como de campo, se ha determinado que en la ciudad de Ambato, un gran número de motocicletas no cuentan con algún tipo de sistema de rastreo o de verificación de usuario.

Actualmente grandes marcas como BMW en su línea de motocicletas Motorrad, incorpora la tecnología “Navigator” basada en la utilización de un GPS a bordo con almacenamiento 8 GB para mapas descargados. Se ofrece además HMI (*Human to Machine Interface*, Interfaz Hombre Máquina) para manejar una amplia gama de controles de la motocicleta. En lo que se refiere a seguridad, el bloqueo de la motocicleta activa la memoria de posicionamiento para activar una alarma en caso de manipulación o movimiento no autorizado. Los sistemas de conducción asistida utilizados para el modelo R1200 incluyen asistencia electrónica, con la cual se puede cambiar marchas sin necesidad de utilizar el embrague o el acelerador [48]. Harley Davidson ofrece soluciones similares con sistemas de navegación GPS en sus modelos de gama Alta Road Glide y Road Glide Special [49]. KTM con su línea Adventure, incorpora asistencia de manejo para el control de la velocidad y la tracción, además de asistencia de frenado para el sistema ABS (*Anti Block System*, Sistema Anti Bloqueo) [50].



Todos los fabricantes mencionados con anterioridad ofrecen modelos costosos disponibles desde 10000 €. Este tipo motocicletas de bella estética y alta funcionalidad no puede ser adquirido por la clase media a nivel nacional. En pos de este hecho, las ensambladoras nacionales de modelos de origen chino, tales como Shineray y Motor Uno, ofrecen una amplia gama de modelos disponibles desde USD 1200. La economía de sus modelos, ha provocado que su mercado se amplíe notablemente entre los consumidores a nivel nacional. Pese a que la calidad de los componentes mecánicos incorporados por estos modelos es bastante buena en comparación a modelos de marcas de renombre, la electrónica de los mismos es bastante limitada. En lo que a seguridad concierne, la posesión de la llave del vehículo basta para la sustracción del mismo. No se ofrecen modelos con sistemas de rastreo o alarmas, lo cual convierte a las motocicletas ensambladas a nivel nacional en un blanco fácil de la delincuencia.

El auge de y la expansión de los sistemas microcontroladores, permiten digitalizar e intervenir sistemas para automatizar o potenciar la operación de los mismos. Es posible y factible tanto económica como técnicamente desarrollar un sistema prototipo que incluya verificación de usuario, junto a un sistema de rastreo y de conducción asistida para las motocicletas ensambladas a nivel nacional. De esta forma se brindan al vehículo un alto valor agregado, que protege la inversión realizada en la adquisición del mismo.

#### **4.2. DIAGRAMA DE BLOQUES DEL SISTEMA**

Como primera instancia, se procede a detallar los bloques respectivos que conformarán el sistema electrónico de control de seguridad física y conducción asistida de una motocicleta. El estudio individual de cada una de las etapas conlleva como propósito determinar las características óptimas de cada uno de los componentes así como su disposición para la implementación posterior de un prototipo adecuado que satisfaga los objetivos y condiciones iniciales del proyecto. Los bloques del sistema son detallados en la figura 4.1.

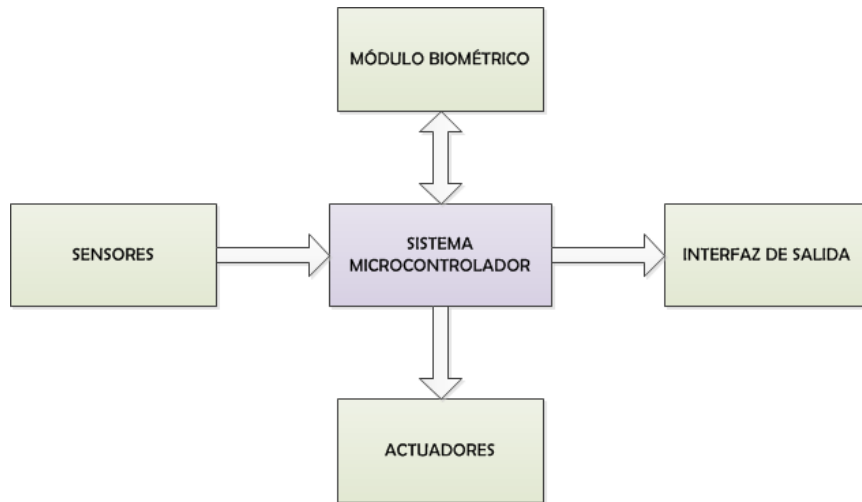


Figura 4.1: Diagrama de bloques del sistema  
**Elaborado por:** El Investigador.

- ▶ Sistema Microcontrolador: el bloque principal del sistema se encarga del procesamiento de la información recogida por parte de los sensores, de la verificación de datos adquiridos por el sistema biométrico, la ejecución de acciones reflejadas en los actuadores y la comunicación con el usuario mediante las interfaces de salida, todo, mediante la toma de decisiones estructuradas sobre una programación adecuada. Capacidad de procesamiento adecuada, bajo coste, universalidad de programación y características robustas tanto física como eléctricamente, convierten a los sistemas microcontrolados en alternativas ideales para el desarrollo de proyectos de sistemas embebidos.
- ▶ Sensores: la recolección de información sobre variables físicas del mundo real para su posterior transducción en señales eléctricas listas para ser procesadas digitalmente es un proceso fundamental para cualquier sistema de control.
- ▶ Actuadores: debido a que los sistemas microcontrolados operan con valores de tensión y corriente pequeños, por alrededor de 5V DC, es necesaria la amplificación y conmutación de las señales generadas por los mismos.
- ▶ Interfaz de salida: para completar el diseño y mantener al usuario adecuada y oportunamente informado sobre las acciones y ejecuciones del sistema, se requiere de una interfaz de salida que se acople a las condiciones sobre las cuales actúa el sistema.

### **4.3. CRITERIOS DE SELECCIÓN DE EQUIPOS**

La selección de equipos para el sistema electrónico de control de seguridad física y conducción asistida de una motocicleta se llevará a cabo tomando en cuenta 3 criterios indispensables: técnico, económico, y geográfico.

- ▶ Técnico: el sistema de control de seguridad física requiere equipos que cumplan las exigencias propias del sistema. Adicionalmente los equipos seleccionados deben ser capaces no únicamente de llevar a cabo todas las operaciones y procesos embebidos tanto del sistema de control de seguridad físico como el de conducción asistida; sino que también deben realizarlo con una velocidad de reloj óptima y con un consumo energético mínimo en la medida de lo posible.
- ▶ Económico: el capital invertido para la selección de equipos debe ser relativamente pequeño en comparación al modelo de pruebas. Caso contrario, se excede el límite de adquisición de la clase baja - media, a la cual está dirigida esta sección de vehículos.
- ▶ Geográfico: los equipos seleccionados deben estar disponibles a nivel nacional para la implementación oportuna dentro de los plazos estipulados del proyecto.






Todos los equipos seleccionados para las funciones de cada uno de los componentes del diagrama de bloques serán escogidos mediante la aplicación conjunta de los criterios anteriormente mencionados.

### **4.4. SELECCIÓN DEL SISTEMA MICROCONTROLADOR**

Como elemento principal de procesamiento del sistema, el mismo debe estar basado en arquitectura ARM, con una velocidad de procesamiento superior a los 4 MHz. Se ha fijado esa velocidad de reloj porque es la mínima adecuada para la implementación de las operaciones y procesos relacionados al sistema de conducción asistida (debido a la constante inspección de instrumentación y señales de control).

Se han seleccionado 5 placas de hardware libre basadas en arquitectura ARM para la elección de la unidad central de procesamiento, cuyas características principales son resumidas en la tabla 4.1

Tabla 4.1: Características de los sistemas microcontrolados preseleccionados

PLACA	BEAGLEBONE REV A5	RASPBERRY PI B2	RASPBERRY PI A+	ARDUINO MEGA 2560 R3	PAPILIO ONE 500K
IMAGEN					
DIMENSIONES	86mm x 54mm	85mm x 56mm	65mm x 56mm	102mm x 54 mm	68mm x 68mm
PROCESADOR	ARM-CORTEX A8	ARM-CORTEX A7	ARM1176JZFS	ATMEGA 2560	Xilinx Spartan 3E FPGA
LONGITUD DE PALABRA	32 bits	32 bits	32 bits	8 bits	8 bits
VELOCIDAD DE RELOJ	700 MHz	900 MHz	700 MHz	16 MHz	32 MHz, variable
RAM	256 MB DDR2	1 GB DDR2	256 MB DDR2	8 KB	8 KB
VOLTAJE DE ALIMENTACIÓN	5 V	5 V	5 V	7 -12 V	6.5 – 10 v
GPIO	65	8	40	54	48
CORRIENTE POR GPIO	12 mA	<10 mA	15 mA	40mA	32mA
RESOLUCIÓN DE ENTRADAS ANALÓGICAS	7 – 12 bits	NA	6 – 10 bits	6 – 10 bits	6-10 bits
UART	5	1	2	4	2
DOCK LANDING	NA	SI	SI	SI	NO
PROGRAMACIÓN NATIVA	PYTHON	PYTHON	PYTHON	ARDUINO (basado en C++)	Trasladable a Arduino IDE
PROGRAMACIÓN FLEXIBLE	SI	SI	SI	SI	NA
DISPONIBILIDAD	EXTERNA	LOCAL	EXTERNA	LOCAL	EXTERNA
PRECIO	USD 89	USD 92	USD 32.90	USD 38	USD 65
<b>Elaborado por:</b> El Investigador, en base a [51, 52, 53, 54, 55].					

Del análisis de la tabla 4.1, se determinan las siguientes conclusiones:

- ▶ Beagle Bone ofrece una capacidad de procesamiento que le permite ejecutar versiones de Linux con suma fluidez. Su precio es elevado y lamentablemente no está disponible a nivel nacional. Existe actualmente el modelo Black, que es físicamente muy parecido a la placa original, con la diferencia en que es más económico, justificándose en su rendimiento.
- ▶ Raspberry surgió como un ordenador básico con todas las prestaciones y al mínimo de precio. Pero conforme han ido evolucionando sus modelos originales, el A y el B, se han ido diferenciando y adaptándose a la competencia. Actualmente el modelo A+ está enfocado a su uso en sistemas embebidos para hacer frente a Arduino. Se ha disminuido la potencia de procesamiento así también como se han aumentado el número de pines digitales de propósito general. La poca corriente que puede ingresar o se drenada de cada uno de los pines digitales ( $\cong 15 mA$ ), la falta de circuitos de reseteo, la carencia de elementos de protección contra corriente y la alimentación restringida a valores nominales de 5V en continua; resultan en que no sean prácticos a la hora de interactuar con elementos presentes en varios sistemas embebidos.
- ▶ Arduino debido a sus prestaciones, robustez, compatibilidad, flexibilidad de entornos de programación, circuitos de protección y amplio rango de voltajes de alimentación, viene a ser una de las mejores opciones al momento de desarrollar sistemas embebidos.
- ▶ Papilio One 500K, es una placa muy poco conocida basada en Arduino pero con una características bastantes interesantes como la inclusión de un cristal de 32 MHz que puede ser configurado mediante hardware para trabajar a frecuencias mayores o menores. Pese a estar basada en uno de los proyectos más populares de Hardware libre, poco se conoce sobre su compatibilidad con módulos adicionales desarrollados para su plataforma de derivación.

#### **4.4.1. EVALUACIÓN DE LOS SISTEMAS MICROCONTROLADOS**

Los criterios de calificación para la elección de la placa del sistema micro controlado son los siguientes:

- ▶ Velocidad de reloj: a mayor velocidad de reloj, mayor puntuación.
- ▶ Voltaje de Alimentación: a menor voltaje de alimentación, mayor puntuación.
- ▶ Número de GPIO's: un mayor número de pines de propósito general conllevarán a una mayor puntuación.
- ▶ Compatibilidad con módulos fabricados por terceros: a mayor número de módulos compatibles fabricados por terceros, mayor puntuación.
- ▶ Número de puertos UART: a mayor número de interfaces universales asincrónicas transceptoras, mayor puntuación.
- ▶ Coste: a menor coste, mayor puntuación.
- ▶ Disponibilidad: una disponibilidad local e inmediata resultará en una mayor puntuación.

Todos los criterios de calificación mencionados con antelación, estarán valorados en una escala de calificación de 1 a 3. La evaluación de los sistemas microcontroladores se puede apreciar en la tabla 4.2.

Tabla 4.2: Calificación de parámetros de selección para el sistema microcontrolador

<b>PLACA</b>	<b>BEAGLEBONE REV 5A</b>	<b>RASPBERR Y PI B2</b>	<b>RASPBERR Y PI A+</b>	<b>ARDUINO MEGA 2560 R3</b>	<b>PAPILIO ONE 500K</b>
<i>VELOCIDAD DE RELOJ</i>	3	3	3	1	1
<i>VOLTAJE DE ALIMENTACIÓN</i>	1	1	1	3	3
<i>NÚMERO DE GPIO'S</i>	3	2	3	3	3
<i>COMPATIBILIDAD</i>	2	2	2	3	1
<i>PUERTOS UART</i>	3	2	2	3	1
<i>DISPONIBILIDAD</i>	1	3	1	3	1
<i>COSTE</i>	1	1	2	3	2
<i>TOTAL</i>	14	14	14	19	12
<i>PROMEDIO</i>	2	2	2	<b>2,71</b>	1,71
<b>Elaborado por:</b> El Investigador.					

En base a los resultados de la tabla, se selecciona a Arduino como la plataforma de desarrollo para el sistema electrónico de control de seguridad física y conducción asistida de una motocicleta.

#### 4.4.2. ARDUINO MEGA 2560 R3

Si bien es cierto que la placa UNO, es la más popular a nivel mundial del proyecto Arduino, MEGA goza también de amplia aceptación. Más pines, más capacidad de procesamiento y el mismo nivel lógico de procesamiento que UNO, hacen que MEGA sea la primera opción a considerar en el desarrollo de sistemas embebidos que requieran controlar una gran cantidad de dispositivos.

La principal diferencia de MEGA2560 con su primera versión, está localizada en el microcontrolador, ya que originalmente utilizaba el ATMEGA1280. Entre otras características íntimamente relacionadas al desempeño del micro, el ATMEGA2560, provee de un circuito de reseteo más estable para la placa. El aspecto físico de la placa se aprecia en la figura 4.2.



Figura 4.2: Arduino Mega 2560 R3

**Fuente:** Arduino Mega [56].

##### 4.4.2.1. CARACTERÍSTICAS ELÉCTRICAS ARDUINO MEGA 2560 R3

Se detalla el resumen general de características eléctricas de Arduino MEGA 2560 R3 en la tabla 4.3.

Tabla 4.3: Características Eléctricas Arduino Mega 2560 R3

<b>Microcontrolador</b>	ATmega2560
<b>Voltaje de Alimentación</b>	5V
<b>Voltaje de Entrada (recomendado)</b>	7-12V
<b>Límite de Voltaje de Entrada</b>	6-20V
<b>Pines Digitales E/S</b>	54
<b>Pines PWM</b>	15
<b>Entradas Analógicas</b>	16
<b>Corriente de salida para pines digitales</b>	40 mA
<b>Memoria Flash</b>	256 KB (8 KB reservados)
<b>SRAM</b>	8 KB
<b>EEPROM</b>	4 KB
<b>Velocidad del Reloj</b>	16 MHz
<b>Fuente:</b> Arduino Mega [57].	

Mega, a diferencia de otras placas ofrece 4 puertos seriales de lógica TTL para interactuar con dispositivos que manejen comunicación serial.

- ▶ Serial 0: pin 0 como trasmisor, pin 1 como receptor.
- ▶ Serial 1: pin 19 como trasmisor, pin 18 como receptor.
- ▶ Serial 2: pin 17 como trasmisor, pin 16 como receptor.
- ▶ Serial 3: pin 15 como trasmisor, pin 14 como receptor.

Adicionalmente, la librería <SoftwareSerial.h>, permite configurar una variedad de pines digitales como receptores y transmisores de puertos seriales personales.

Para un detalle ampliado de la placa Arduino Mega, revisar el anexo B1.

#### 4.4.3. ARDUINO UNO SMD R3

Ésta es la tercera revisión de una de las placas más populares del proyecto de Hardware Libre Arduino. El módulo Uno SMD (*Surface Mount Device*, Dispositivo de Montaje Superficial) es una placa con microcontrolador integrado basada en el modelo ATMEGA328. En la figura 4.3 se puede apreciar el aspecto físico de la placa.





Figura 4.3: Arduino Uno SMD R3  
**Fuente:** Arduino Uno SMD [58].

La diferencia con el modelo Uno estándar, reside en el tipo de montaje del microcontrolador incorporado. La placa estándar posee la versión tipo DIL (*Dual-in-Line*, Doble Línea), mientras que ésta versión utiliza el tipo SMD.

#### 4.4.3.1. CARACTERÍSTICA ELÉCTRICAS ARDUINO UNO SMD R3

Se detalla el resumen general de características eléctricas de Arduino Uno SMD R3 en la tabla 4.4.

Tabla 4.4: Características Eléctricas Arduino Uno SMD R3

<b>Microcontrolador</b>	ATMEGA328
<b>Voltaje de Alimentación</b>	5V
<b>Voltaje de Entrada (recomendado)</b>	7-12V
<b>Límite de Voltaje de Entrada</b>	6-20V
<b>Pines Digitales E/S</b>	20
<b>Pines PWM</b>	6
<b>Entradas Analógicas</b>	6
<b>Corriente de salida para pines digitales</b>	40 mA
<b>Memoria Flash</b>	32 KB (0.5 KB reservados)
<b>SRAM</b>	2KB
<b>EEPROM</b>	1 KB
<b>Velocidad del Reloj</b>	16 MHz
<b>Fuente:</b> Arduino Uno SMD [59].	



Esta placa dispone de 20 pines digitales configurables tanto como entradas como salidas. La lógica de operación es TTL, operando a 5 V. Cada pin puede soportar o proporcionar 40 mA de corriente. Los puertos 0 y 1 son utilizados para transmisión y recepción serial de datos en caso de añadir shields (placas diseñadas por terceros totalmente compatibles con Arduino) específicos para añadir funcionalidades nuevas no soportadas por la placa original (conectividad GSM, por ejemplo).

Para un detalle ampliado de la placa Arduino Uno, revisar el anexo B2.

#### 4.5. SELECCIÓN DEL MÓDULO GSM/GPS

Las notificaciones y comandos relacionados al sistema de control de seguridad física, deben viajar preferentemente por la red celular, con el fin de asegurar compatibilidad con cualquier tipo de teléfono celular que disponga el usuario final. Se han seleccionado los módulos que se aprecian en la tabla 4.5.

Tabla 4.5: Características de módulos GSM/GSPS preseleccionados

MÓDULO GSM/GPS	SIMCOM 908 KITS UNO	GSM GPS PLAYGROUND SHIELD
IMAGEN		
BANDA	Cuádruple Clase 10, Estación móvil clase B	Cuádruple Clase 10, Estación móvil clase B
GSM	X	X
GPRS	X	X
GPS	X	X
IDE NATIVO	COMANDOS AT	PYTHON
ADICIONALES	Display LCD 5110	Detección de Jamming
COMPATIBLE CON ARDUINO	X	X
DISPONIBILIDAD	LOCAL	EXTERNA
PRECIO	USD 65	USD 129.95
<b>Elaborado por:</b> El Investigador en base a [60, 61].		

El módulo SIMCOM 908, ofrece conectividad GSM, GPRS Y GPS en una sola placa. Puede ser programado a través del IDE de Arduino, pero también se puede usarlo independientemente mediante su programación nativa en comandos AT (AT: Atención). Un valor agregado a su atractivo precio es la inclusión de un display LCD 5110 que cuenta con su propio puerto en la placa.

GSM GPS PLAYGROUND SHIELD incluye programación nativa en Python. Posee detección de jamming, lo cual lo vuelve una opción bastante interesante cuando se trate de desarrollar proyectos en los que se intenta vulnerar la seguridad o cobertura de una red celular. Precisamente la inclusión de esta innovación justifica su coste, que es elevado en comparación a otros módulos disponibles en el mercado.

#### **4.5.1. EVALUACIÓN DE LOS MÓDULOS GSM/GPS**

Los criterios de calificación para la elección del módulo GSM/GPS son los siguientes:

- ▶ Banda: a mayor número de bandas compatibles, mayor puntuación.
- ▶ Capacidad de overclocking: se entrega una puntuación máxima en caso de que el sistema posea capacidad de overclocking. De no poseerla, la puntuación que se asignará será mínima.
- ▶ Extras incorporados: a mayor número de aditamentos u opciones de desarrollo incorporadas, mayor puntuación.
- ▶ Compatibilidad con Arduino: si el módulo posee una versión tipo shield para Arduino, la puntuación entregada será máxima. De ser un módulo compatible mediante cualquier otro método de comunicación, la puntuación será mínima.
- ▶ Coste: a menor coste, mayor puntuación.
- ▶ Disponibilidad: una disponibilidad local e inmediata resultará en una mayor puntuación.

Todos los criterios de calificación mencionados con antelación, estarán valorados en una escala de calificación de 1 a 3. La evaluación de los módulos GSM/GPS se puede apreciar en la tabla 4.6.

Tabla 4.6: Calificación de parámetros de selección para el módulo GSM/GPS

<b>PLACA</b>	<b>SIM908 KITS UNO</b>	<b>GSM GPS PLAYGROUND SHIELD</b>
<i>BANDA</i>	3	3
<i>OVERCLOCKING</i>	3	1
<i>EXTRAS</i>	2	3
<i>COMPATIBILIDAD</i>	3	3
<i>DISPONIBILIDAD</i>	3	1
<i>COSTE</i>	2	1
<i>TOTAL</i>	16	12
<i>PROMEDIO</i>	<b>2.66</b>	2
<b>Elaborado por:</b> El Investigador.		

Debido a que ambos módulos poseen características bastante similares, se opta por realizar la elección de la placa SIMCOM 908, debido a la disponibilidad a nivel nacional y a que su coste se ajusta a las necesidades del desarrollo del prototipo del sistema.

#### 4.6. SIM908 KITS UNO

SIMCOM 908 es un módulo cuatribanda GSM/GPRS que combina tecnología GPS para incluir funciones de navegación a cualquier sistema electrónico. Este shield (*shield*, módulo) compacto ciertamente ahorro costos e inconvenientes para aquellos que buscan la movilidad de la telefonía móvil con la posibilidad de utilizar funciones GPS. Este módulo puede rastrear múltiples elementos a la vez a cualquier instante y en cualquier ubicación. En la figura 4.4 se puede apreciar la placa del módulo descrito a la vez que sus características principales son detalladas en la tabla 4.7.



Figura 4.4: Módulo SIM 908, SIMCOM

**Fuente:** GSM GPS shield for UNO SIM908 developments kits [62].

Tabla 4.7: Resumen General SIMCOM908

<b>Tipo</b>	<b>Características</b>
<i>General</i>	<ul style="list-style-type: none"> <li>▪ 4 bandas (850, 900, 1800, 1900) MHz</li> <li>▪ GPRS multiclúster de clase 10</li> <li>▪ Estación tipo B para GPRS</li> <li>▪ Comandos AT</li> <li>▪ Bajo consumo de energía</li> </ul>
<i>SMS mediante GSM o GPRS</i>	<ul style="list-style-type: none"> <li>▪ Punto a Punto</li> <li>▪ Broadcast de célula</li> <li>▪ Modo PDU</li> </ul>
<i>GPS</i>	<ul style="list-style-type: none"> <li>▪ Recibidor de 43 canales</li> <li>▪ Arranque en frío 30s.</li> <li>▪ Arranque en caliente 1s.</li> <li>▪ Error menor a 2.5m en posición horizontal</li> </ul>
<b>Fuente:</b> SIM908 Datasheet [63].	

SIM908 está basado en ARM, específicamente, ARM926EJ-S que dobla la frecuencia de procesamiento del núcleo, comparado a otros módulos basados en ARM7. Las mejoras incluidas en la función “sleep”, hace que el consumo de energía sea hasta un 40% más bajo comparado con el de otras opciones similares en el mercado.

Para un detalle ampliado del circuito integrado SIM908, revisar el anexo B3.

Los módulos basados en el integrado SIMCOM 908 poseen un led indicador de estado, mismo que notifica sobre el estado de funcionamiento del módulo y su interacción con la red. La tabla 4.8 detalla los posibles comportamientos del led de notificación incluido.

Tabla 4.8: Estados del LED indicador SIMCOM 908

<b>Estado</b>	<b>Descripción</b>
<b>Apagado</b>	El módulo no está en funcionamiento
<b>64 ms alto / 800 ms bajo</b>	El módulo no puede conectarse a la red
<b>64 ms alto / 3000 ms bajo</b>	El módulo se ha conectado a la red
<b>Fuente:</b> SIM908 Hardware Design [64].	

#### **4.6.1. PROGRAMACIÓN SIMCOM 908**

Todos los módulos de la familia SIMCOM, pueden ser programados mediante dos métodos principales.

- ▶ Comandos HAYES
- ▶ Librerías compatibles

A continuación, se detalla una breve descripción, así como las ventajas y desventajas que ofrecen cada una de estas formas de programación.

##### **4.6.1.1. COMANDOS HAYES PARA SIMCOM 908**

Es posible realizar la programación de SIMCOM 908 mediante comandos AT, desarrollados por HAYES Communications.

La normativa ITU-T V.250, sobre Marcado Serial Asíncrono Automático, define las parametrizaciones de los comandos AT básicos.

##### **4.6.1.2. LIBRERÍAS COMPATIBLES CON SIMCOM 908**

La sociedad del hardware libre trabaja continuamente y en colaboración con otros entusiastas para desarrollar librerías compatibles para los módulos o elementos electrónicos desarrollador por terceros para Arduino. Estas librerías pueden ser utilizadas incluso en sistemas micro controlados más simples como AVR's o plataformas de desarrollo con características similares a las de Arduino.

Open Electronics, pone a disposición la librería GSMShield, totalmente compatible con los módulos SIMCOM, con funciones que facilitan y optimizan la programación de los mismos en el IDE de Arduino.



#### **4.7. SELECCIÓN DEL SISTEMA BIOMÉTRICO**

Elemento indispensable a utilizarse en el sistema de control de seguridad física. El mismo debe cumplir con las siguientes condiciones:

- ▶ Tipo no invasivo
- ▶ Alto índice de seguridad
- ▶ Índice de permanencia aceptable
- ▶ Características de comunicación, alimentación y de procesamiento afines a las de la unidad central.

En base a los parámetros de selección previamente detallados, se han seleccionado dos placas orientadas a la verificación de identidad mediante biometría de características no invasivas con el usuario cuyas características principales se aprecian en la tabla 4.9.

Tabla 4.9: Características de los módulos biométricos preseleccionados

MÓDULO	EASY VR 2.0	Roboard RMG-212
<b>IMAGEN</b>		
<b>TECNOLOGÍA</b>	Reconocimiento de Voz	Térmica
<b>SEGURIDAD</b>	Media	Media
<b>PERMANENCIA</b>	Media	Media
<b>DESCRIPCIÓN</b>	Módulo digitalizador de voz con EEPROM	Matriz de 16x4 de termo sensores
<b>COMUNICACIÓN</b>	Serial	I2C
<b>ADICIONALES</b>	Sintetizador de sonidos integrado	NA
<b>COMPATIBLE CON ARDUINO</b>	X	X
<b>DISPONIBILIDAD</b>	LOCAL	EXTERNA
<b>PRECIO</b>	USD 60	USD 45
<b>Elaborado por:</b> El Investigador, en base a [65, 66].		

EasyVR es proyecto desarrollado por VEEAR Electronics. A simple vista no es diferente de algún simple módulo de voz, nada más lejos de la realidad. Posee un juego de instrucciones pre-programadas disponibles de más de 28 idiomas y un set especial de comandos tipo contraseña que los vuelve ideal para aplicaciones biométricas. Además incluye un ecualizador con el cual se pueden reproducir tablas de sonido previamente cargadas a la memoria de la placa. Existen dos modelos:

- ▶ Placa original: compatible con cualquier microcontrolador mediante comunicación serial.
- ▶ Shield Arduino: diseñado específicamente para acoplarse al pinout de una placa Arduino y utilizar el UART integrado de los pines 0 Y 1.

Roboard RMG-212 es una placa basada en el sensor térmico de tipo matriz RMG-212. El reconocimiento de patrones térmicos, sobre todo faciales, constituye un método biométrico de verificación eficiente y poco invasivo. Su precio ciertamente lo convierte en una alternativa muy interesante. Sin embargo, la utilización de I2C como protocolo de comunicación complica notablemente el desarrollo de algún prototipo en el que se requieran elementos de comunicación adicionales. Pese a que Arduino soporta I2C, la configuración un protocolo de comunicación adicional puede ralentizar la velocidad de ejecución de los programas cargados en las placas.

#### **4.7.1. EVALUACIÓN DE LOS SISTEMAS BIOMÉTRICOS**

Los criterios de calificación para la elección del sistema biométrico son los siguientes:

- ▶ Procesador: la presencia de un procesador en el módulo será calificada con una puntuación misma. En caso de ser un sistema basado únicamente en sensores, la puntuación será mínima.
- ▶ Permanencia de la variable biométrica: a mayor permanencia de la variable biométrica utilizada, mayor puntuación.
- ▶ Inmunidad de la variable biométrica: a mayor inmunidad de la variable biométrica utilizada, mayor puntuación.
- ▶ Extras incorporados: a mayor número de aditamentos u opciones de desarrollo incorporadas, mayor puntuación.
- ▶ Compatibilidad con Arduino: si el módulo posee una versión tipo shield para Arduino, la puntuación entregada será máxima. De ser un módulo compatible mediante cualquier otro método de comunicación, la puntuación será mínima.
- ▶ Coste: a menor coste, mayor puntuación.
- ▶ Disponibilidad: una disponibilidad local e inmediata resultará en una mayor puntuación.

Todos los criterios de calificación mencionados con antelación, estarán valorados en una escala de calificación de 1 a 3. La evaluación de los módulos biométricos se puede apreciar en la tabla 4.10.



Tabla 4.10: Calificación de parámetros de selección para el módulo biométrico

<b>PLACA</b>	<b>EASY VR 2.0</b>	<b>ROBOARD</b>
<i>PROCESADOR</i>	3	1
<i>PERMANENCIA DE LA VARIABLE</i>	2	2
<i>INMUNIDAD DE LA VARIABLE</i>	3	1
<i>EXTRAS</i>	3	2
<i>COMPATIBILIDAD</i>	3	3
<i>DISPONIBILIDAD</i>	3	1
<i>COSTE</i>	1	3
<i>TOTAL</i>	18	13
<i>PROMEDIO</i>	<b>2.57</b>	1.85
<b>Elaborado por: El Investigador.</b>		

Debido a que una motocicleta es un vehículo en el cual el piloto está expuesto de forma notable a las condiciones ambientales, en algunos casos como frío o calor intenso, la variable de entrada del sistema biométrico debe poseer cierto grado de inmunidad a las condiciones ambientales. Ya que los cambios de temperatura extremos pueden influir temporalmente sobre los patrones de temperatura de una imagen térmica, se descarta el uso de Roboard. La voz en cambio, no se ve afectada por este tipo de factores.

Por su versatilidad, y la inclusión de un sintetizador, mismo que puede ser utilizado como apoyo a la interfaz de salida del prototipo de sistema electrónico de control de seguridad física y conducción asistida de una motocicleta; se elige a EASY VR 2.0 como el módulo de reconocimiento biométrico, que conformará el bloque de seguridad del sistema.

#### **4.8. EASY VR SHIELD 2.0**

EasyVR es un robusto módulo de reconocimiento de voz. Su simplicidad y versatilidad, lo hacen adaptable a cualquier tipo de proyectos.

El módulo ya incluye sets de instrucciones pre programadas, listos para ser utilizados. El número total de instrucciones configurables es de 32. No está demás mencionar que los comandos personalizados pueden estar en cualquier idioma. Sin embargo, si se quieren utilizar comandos en algún lenguaje en específico, el firmware incluye 28 idiomas por defecto.

En la figura 4.5 se aprecia el módulo en su versión shield para Arduino.

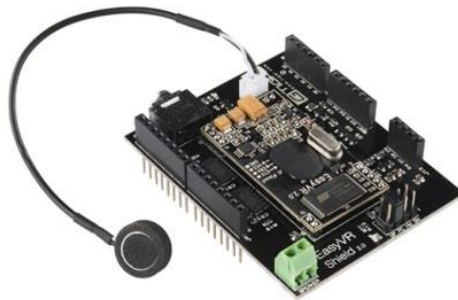


Figura 4.5: EasyVR shield 2.0 para Arduino  
**Fuente:** EasyVR User Manual [67].

#### 4.8.1. CARACTERÍSTICAS ELÉCTRICAS EASY VR 2.0

Se muestra a continuación en la tabla 4.11, un resumen general de las características eléctricas de esta placa.

Tabla 4.11: Características eléctricas EasyVR Shield 2.0

Símbolo	Parámetro	Valor
$V_{IH}$	Voltaje alto de entrada	$[2.4 - 3.3]V$
$V_{IL}$	Voltaje bajo de entrada	$[-0.1 - 0.75]V$
$I_{IL}$	Corriente de fuga	$[1 - 10]\mu A$
$R_{PU}$	Resistencia de pull-up	Alta: $10K\Omega$ Baja: $200K\Omega$
$V_{OH}$	Voltaje alto de salida, con $I_{OH} = -5mA$	$2.4V$
$V_{OL}$	Voltaje bajo de salida, con $I_{OL} = 8mA$	$0.6V$
<b>Fuente:</b> EasyVR User Manual [68].		

El anexo B4 provee un detalle ampliado de las características de EasyVR.

Debido a que el micrófono incluido en la placa es un elemento indispensable para el reconocimiento de voz, la tabla 4.12 se detallan las características principales de este componente.

Tabla 4.12: Características piezoeléctricas EasyVR Shield 2.0

<b>Tipo</b>	Omnidireccional
<b>Respuesta</b>	$-38db$
<b>Impedancia de carga</b>	$2.2K\Omega$
<b>Voltaje de Operación</b>	$3.3V$
<b>Rango de respuesta plana</b>	$[100 - 20000]Hz$
<b>Fuente:</b> EasyVR User Manual [69].	

Es posible reemplazar el micrófono, únicamente por uno del mismo tipo. Sin embargo, el fabricante no lo recomienda, debido a que las capacidades de reconocimiento de voz pueden verse parcial o notablemente afectadas.

Este modelo incluye también la función de ecualizador, siendo posible reproducir sonidos como respuesta a comandos, lo cual resulta útil como confirmación de órdenes. La placa incluye una bornera a la que puede ser acoplada directamente un parlante de una impedancia de 8 ohm. El formato de los sonidos que pueden ser almacenados y reproducidos es de tipo WAV, muestreado a 22.1 KHz.

#### 4.8.2. COMUNICACIÓN SERIAL DE EASY VR 2.0

La placa, que es de tipo esclavo, maneja comunicación UART, lo cual quiere decir que puede ser utilizada por cualquier dispositivo que utilice este protocolo de comunicación. Se detallan las características de la comunicación empleada por la placa:

- Tasa de bits: 9600 (por defecto), 19200, 38700, 115200
- Trama de 8 bits, sin paridad, bits de parada y de inicio
- IDLE (estado de inactividad) en Vcc.

#### 4.8.3. CONFIGURACIONES FÍSICAS EASY VR 2.0

Es importante revisar la disposición física de los elementos en la placa, detallada en la figura 4.6.

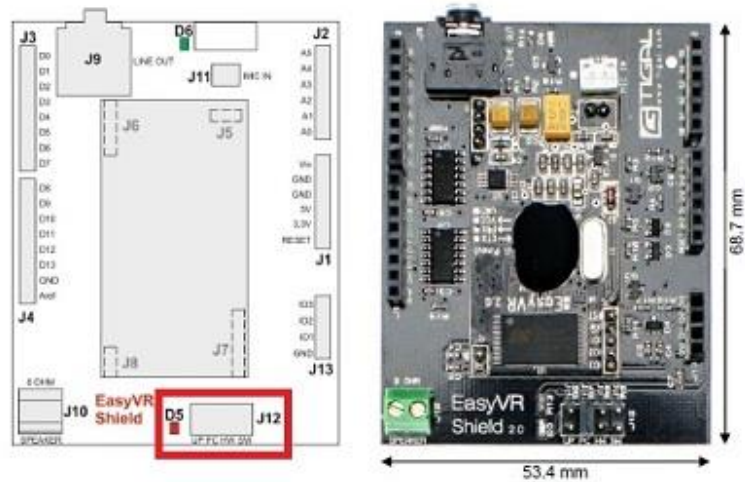


Figura 4.6: Detalle físico EasyVR Shield 2.0

**Fuente:** EasyVR User Manual [70].

La sección J12 que aparece remarcada en la figura 4.6, es un jumper que actúa de multiplexor de 4 modos de operación distintos, cuyas funciones son detalladas en la tabla 4.13, los cuales son necesarios tomar en cuenta para poder trabajar con la placa.

Tabla 4.13: Modos de operación de EasyVR Shield

Posición	Descripción
<b>UP</b>	Para actualizar el firmware de la placa o subir a la memoria EEPROM la tabla de sonidos correspondiente. El controlador para Arduino ingresa en un estado permanente de reset. La placa utiliza únicamente comunicación USB.
<b>PC</b>	Para utilizarse con el EasyVR Commander. El controlador para Arduino ingresa en un estado permanente de reset. La placa utiliza únicamente comunicación USB.
<b>HW</b>	Para utilizarse con Arduino mediante Serial0, pines 0 y 1.
<b>SW</b>	Para utilizarse con Arduino mediante los pines 12 y 13, previa a la configuración respectiva de <SoftwareSerial.h> en el código respectivo.
<b>Fuente:</b> EasyVR User Manual [71].	

Para un detalle ampliado de la placa EasyVR 2.0, revisar el anexo B4.

#### **4.8.4. CONSIDERACIONES ADICIONALES EASY VR 2.0**

La placa EasyVR shield, puede trabajar únicamente con dispositivos que manejen una lógica TTL, es decir de 5V. Motivo por el cual, está asegurada la compatibilidad con los siguientes modelos de Arduino.

- ▶ Uno
- ▶ Leonardo
- ▶ Mega

Easy VR no puede trabajar con placas que manejen una tensión de 3.3V (Modelos de Arduino: Diecimilia, Duemilanove, DUE, Pro Nano, LilyPad).


#### **4.9. SELECCIÓN DE LA INTERFAZ DE SALIDA**

La interfaz de salida consistirá en una pantalla de cristal líquido para que el usuario pueda visualizar los datos del sistema de conducción asistida. Las características para la selección de este componente son detalladas a continuación:

- ▶ Bajo consumo de corriente.
- ▶ Compatibilidad con Arduino.
- ▶ Resolución en píxeles.

En base a los parámetros de selección previamente detallados, se han seleccionado dos displays tipo LCD (*Liquid Crystal Display*, Pantalla de cristal Líquido) de características adaptables para la implementación del prototipo, cuyas principales características se pueden apreciar en la tabla 4.14.

Tabla 4.14: Características de los displays preseleccionados

MÓDULO	GENERIC LCD 2X16	NOKIA 5110 LCD DISPLAY
VISTA		
CONTROLADOR	HITACHI	PHILIPS
LÍNEAS UTILIZABLES	2	5
RESOLUCIÓN	2X16	86X48
TASA DE FPS	<10	>135
COMUNICACIÓN	Paralela	Paralela
COMPATIBLE CON ARDUINO	X	X
DISPONIBILIDAD	LOCAL	LOCAL
PRECIO	USD 7.85	USD 9.99
<b>Elaborado por:</b> El Investigador, en base a [72, 73].		

#### 4.9.1. EVALUACIÓN DE LAS INTERFACES DE SALIDA

Los criterios de calificación para la elección de la interfaz de salida son los que se enumeran a continuación:

- ▶ Resolución: a mayor resolución en pixeles disponibles, mayor puntuación.
- ▶ Tiempo de Inicialización: a menor tiempo de inicialización del microcontrolador incorporado, mayor puntuación.
- ▶ FPS (*Frames Per Second*, Cuadros Por Segundo): a mayor tasa de FPS, mayor puntuación.
- ▶ Librerías gráficas: la existencia de librerías con funciones gráficas compatibles resultarán en una puntuación máxima. La ausencia de las mismas, resultará en una puntuación mínima.
- ▶ Compatibilidad con Arduino: si el módulo posee una versión tipo shield para Arduino, la puntuación entregada será máxima. De ser un módulo compatible mediante cualquier otro método de comunicación, la puntuación será mínima.
- ▶ Coste: a menor coste, mayor puntuación.

- ▶ Disponibilidad: una disponibilidad local e inmediata resultará en una mayor puntuación.

Todos los criterios de calificación mencionados con antelación, estarán valorados en una escala de calificación de 1 a 3. La evaluación de las interfaces de salida se puede apreciar en la tabla 4.15.

Tabla 4.15: Calificación de parámetros de selección para interfaces de salida

<i>DISPLAY</i>	GENERIC 2X16	NOKIA 5110
<i>RESOLUCIÓN</i>	1	3
<i>TIEMPO DE INICIALIZACIÓN</i>	1	3
<i>FPS</i>	2	3
<i>LIBRERÍAS GRÁFICAS</i>	1	3
<i>COMPATIBILIDAD</i>	3	3
<i>DISPONIBILIDAD</i>	3	3
<i>COSTE</i>	3	2
<i>TOTAL</i>	14	20
<i>PROMEDIO</i>	2	<b>2.85</b>
<b>Elaborado por: El Investigador.</b>		

Por la superioridad en características del controlador interno, así también como las posibilidad de programación gráfica mediante librerías, se selecciona a Nokia 5110 LCD display, como la interfaz de salida para el desarrollo del prototipo.

#### **4.10. NOKIA 5110 LCD DISPLAY**

Cuenta con una resolución de 86x48 (píxeles monocromáticos), que lo tornan ideal para una variedad de aplicaciones, incluso siendo posible la reproducción de video (con ayuda de una tarjeta SD y estableciendo una tasa de 15 FPS). Resulta ideal para proyectos de instrumentación o similares, ya que cuenta con una buena cantidad de caracteres disponibles, siendo incluso posible dimensionar la el tamaño de la fuente (sin inclusión de librerías adicionales). El controlador interno es un PCD8544 (fabricante: Philips).

Sus funciones estándar son bastante completas y además cuenta con una librería gráfica bastante robusta.

#### 4.10.1. DISPOSICIÓN DE PINES NOKIA 5110 LCD DISPLAY

El LCD5110 cuenta con 8 pines, la comunicación es de tipo serial síncrona. El consumo de corriente es bastante reducido (6 – 7 mA). El aspecto físico del display se muestra en la figura 4.7.

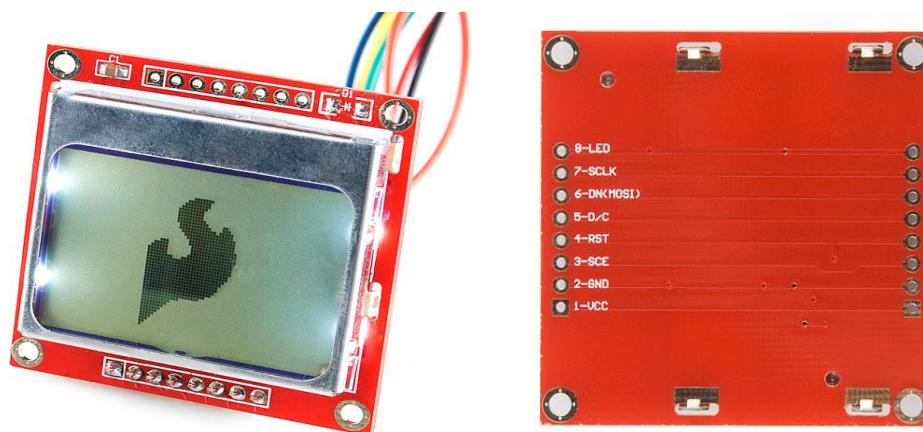


Figura 4.7: Aspecto físico Nokia LCD5110  
**Fuente:** Graphic LCD 84x48 - Nokia 5110 [74].

La tabla 4.16 detalla la descripción de las funciones individuales de cada uno de los pines del display.

Tabla 4.16: Distribución de pines Nokia LCD5110

Pin	Nombre	Función	Características
1	VCC		[2.7-3.3]V
2	GND		
3	SCE	Selección de Chip	Activa Baja
4	RST	Reset	Activa Baja
5	D/C	Selección de Modo	Bajo: modo comando, Alto: modo de datos
6	DN(MOSI)	Data in	
7	SCLK	Serial clock	
8	LED	Backlight del display	Hasta 3.3V

**Fuente:** Nokia LCD 5110 [75].

Como anotación especial, cabe recalcar que en la parte superior derecha de la placa que contiene el display, puede estar o no incluido un diodo. La ausencia o presencia de este



componente no afecta el comportamiento de la pantalla y su inclusión depende meramente del proveedor.

Para un detalle ampliado de la pantalla Nokia LCD 5110, revisar el anexo B5.

#### 4.11. BLOQUE DE SEGURIDAD FÍSICA

El bloque de control de seguridad se basa en la manipulación del switch de corte de la motocicleta, debido a dos características principales de este interruptor:

- ▶ Permite o deniega el arranque del motor mediante encendido eléctrico CDI.
- ▶ Detiene la marcha del motor bajo cualquier instancia en caso de ser desactivado.

En la figura 4.8, se aprecia el aspecto físico de los controles de la mano derecha de una motocicleta, en los cuales se localizan los dos actuadores fundamentales del circuito de ignición eléctrico.


	<b>Part Name</b>	<b>Control Switch RH</b>
	<b>Part No.</b>	<b>DH 2010 49</b>
	<b>Description</b>	The switches operate without any physical contact & thus has no wear & tear of parts. Also these have illumination of tell-tale icons through LED's which glows Bluish-White during night driving.
	<b>Identification Mark</b>	Provision of engine kill switch.

Figura 4.8: Controles de la mano derecha de una motocicleta

**Fuente:** Pulsar DTSi Reparation Manual [76].

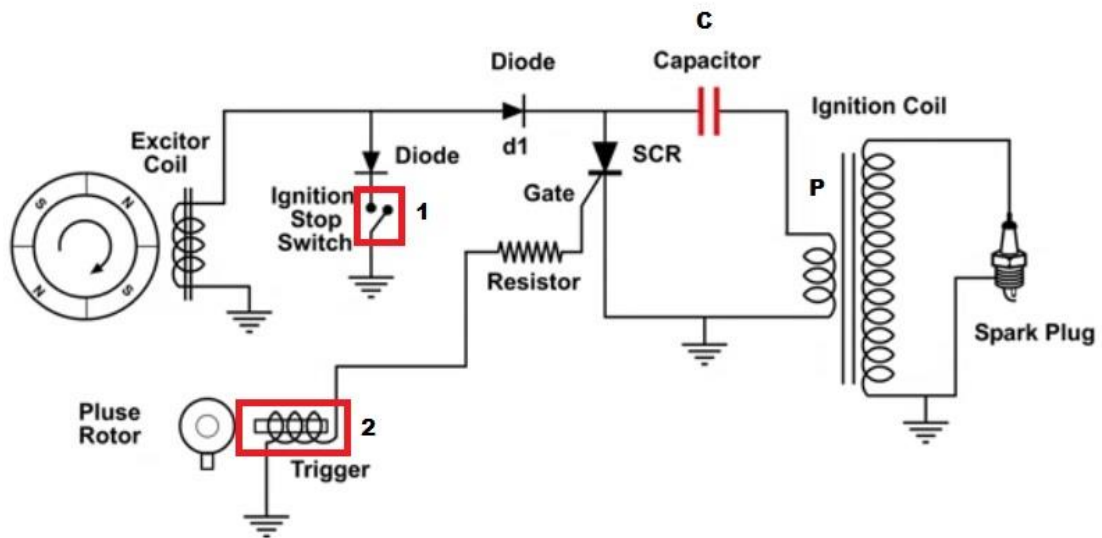
- ▶ Switch de Corte: elemento de control que permite o deniega el arranque eléctrico de la motocicleta, mediante el control de apertura o cierre de la alimentación del circuito de arranque. El mismo puede ser utilizado para apagar el motor con el motor en marcha, mediante su retorno a posición de inactivo. Físicamente es un switch de dos posiciones.
- ▶ Pulsador de arranque: el pulsador de arranque descarga la corriente almacenada por el circuito CDI al motor de arranque para el inicio del motor. Con una batería completamente cargada, la pulsación de este elemento por

aproximadamente 3 segundos logra arrancar el motor. Conforme se pierde la carga en la batería, la pulsación del mismo se acompaña con un ligero jalón del acelerador para poner en marcha al motor. Físicamente en un pulsador de tipo NA sin memoria.

#### 4.11.1. CIRCUITO DE ENCENDIDO POR CDI

No es sino hasta la década de los 80's en el que las motocicletas empiezan a incorporar encendido de tipo eléctrico. Hasta ese entonces, se utilizaba el arranque tradicional mediante patada.

Actualmente el sistema más común y difundido es el CDI (*Capacitor Discharge Induction*, Inducción por Descarga de Condensador), o sistema de descarga por inducción de capacitor, cuyo diagrama esquemático puede ser apreciado en la figura 4.9.



1. Equivalente del switch corte
2. Equivalente del pulsador de ignición.

Figura 4.9: Circuito de encendido mediante descarga por condensador (CDI)

**Fuente:** CDI Capacitor Discharge Ignition Circuit [77].

La diferencia del circuito entre fabricantes es mínima y mantiene constante los elementos principales. El sistema entero se basa en la carga elevada (por alrededor de 300 V) del condensador C. La carga es drenada por completo en la bobina P del primario. Esta aparición elevada y repentina de corriente reflejada por la bobina secundaria de P se dirige hacia la bujía B. La corriente que se genera es más intensa, lo que desencadena en un rendimiento superior de la bujía bajo cualquier régimen.

Las motocicletas con este sistema de encendido cuentan además con un circuito que elige el momento adecuado de la descarga. Un sensor magnético conformado por una bobina y un imán envían la señal a un tiristor, el cual corta el suministro de corriente al condensador C, con la respectiva descarga como efecto, tal y como puede apreciarse en la figura 4.10.

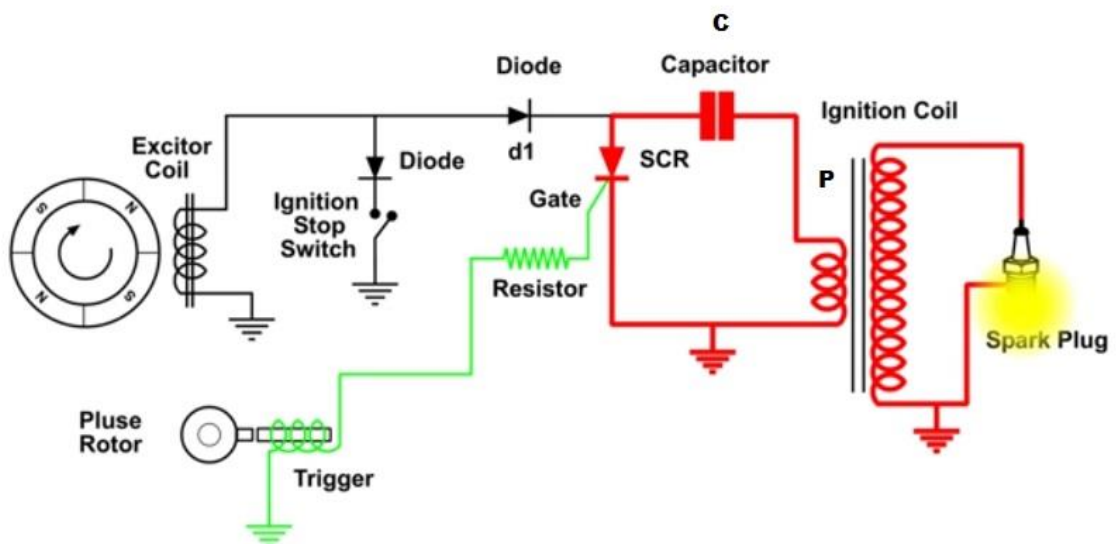


Figura 4.10: Descarga del condensador a través del transformador  
**Fuente:** CDI Capacitor Discharge Ignition Circuit [78].

En la figura 4.11, se aprecia el diagrama eléctrico del sistema de ignición de una motocicleta.

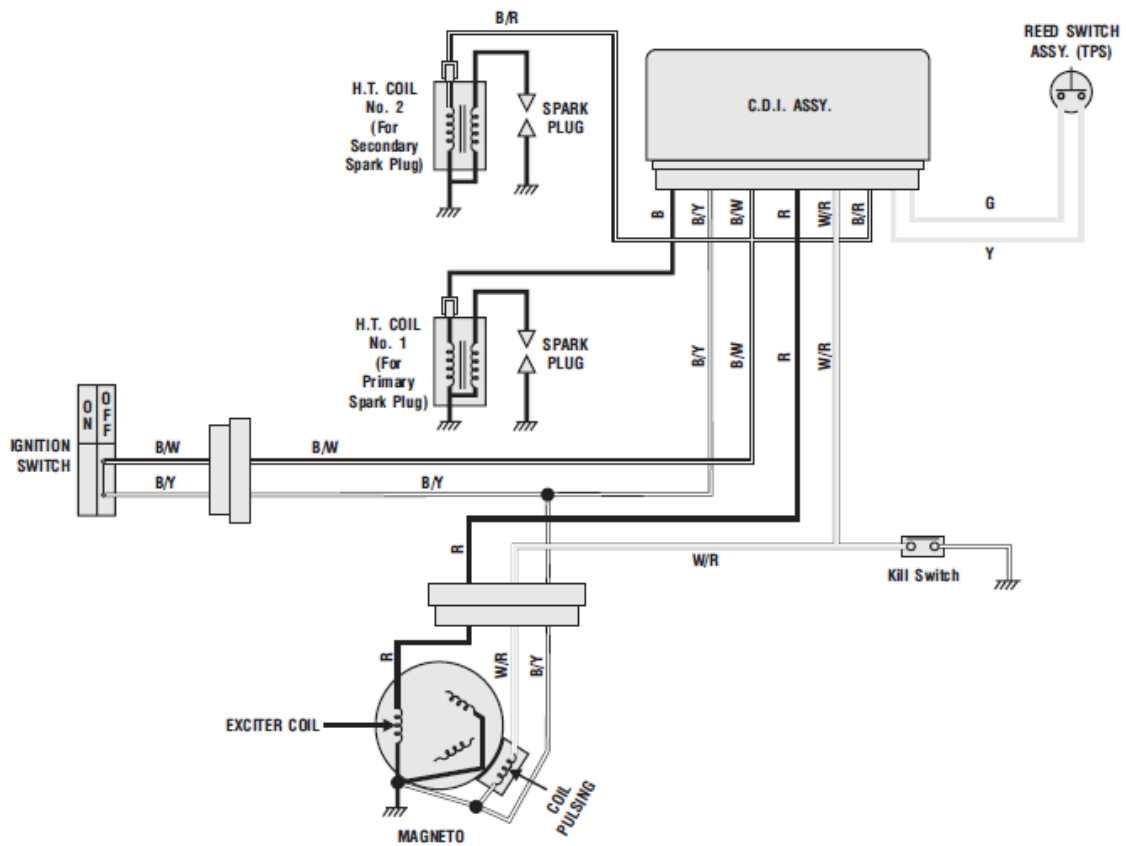


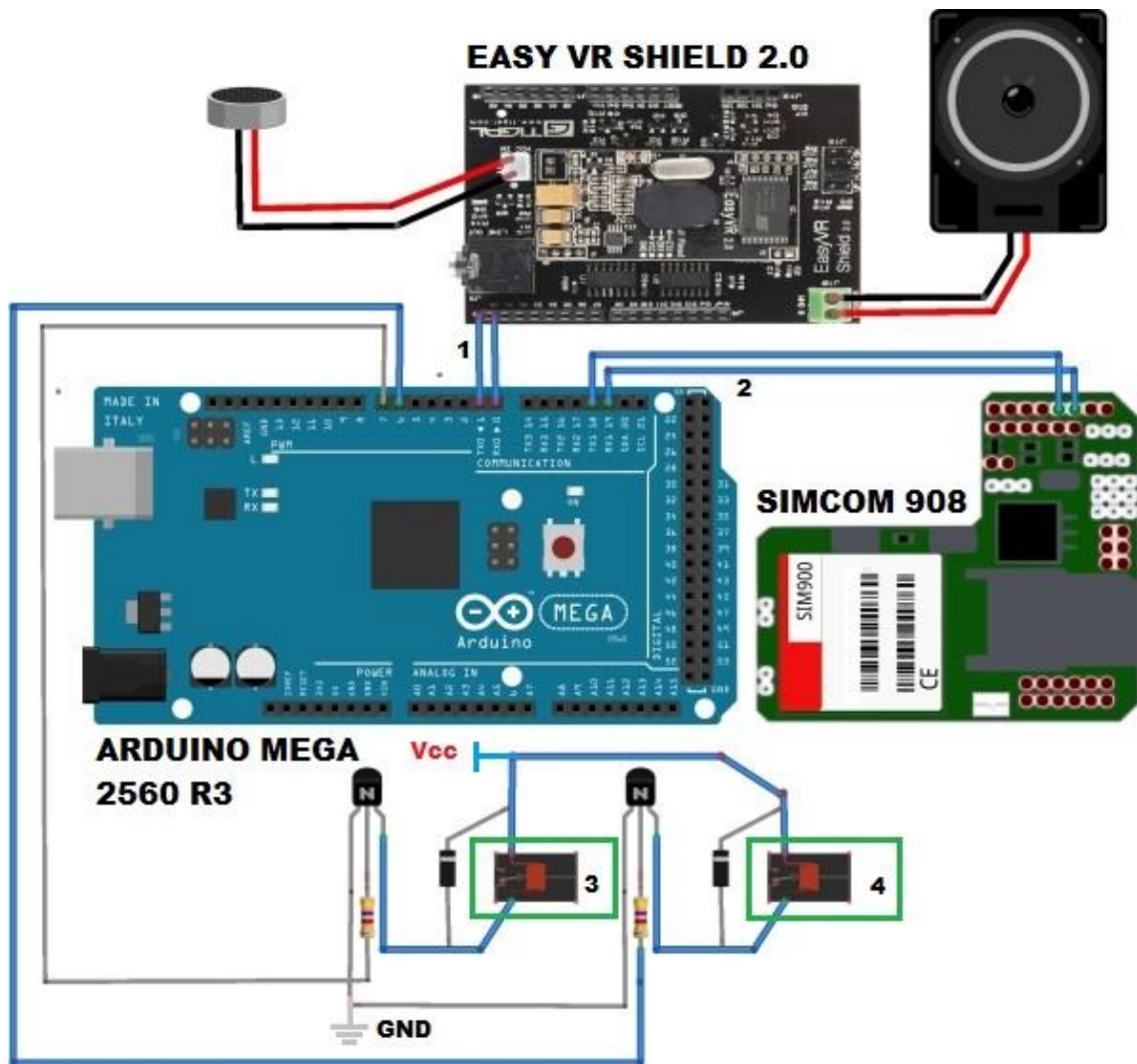
Figura 4.11: Circuito de ignición de una motocicleta.

**Fuente:** Pulsar DTSi Reparation Manual [79].

Tanto el switch de emergencia como el pulsador de arranque serán manipulados mediante salidas digitales con etapa transistorizada.

#### 4.11.2. DISEÑO DEL BLOQUE DE SEGURIDAD FÍSICA

La figura 4.12, detalla el circuito de pruebas para el desarrollo del sistema de control de seguridad.



1. Punteado externo pines TX y RX de SIMCOM con Serial 1.
2. Punteado externo pines TX y RX de Easy VR con Serial 0.
3. Al switch de emergencia.
4. Al pulsador de arranque.

Figura 4.12: Circuito de pruebas bloque de control de seguridad  
**Elaborado por:** El Investigador.

#### 4.11.3. COMANDOS DE VOZ

El bloque de seguridad física cuenta con un módulo de reconocimiento de voz de altas prestaciones. La tabla 4.17 muestra los comandos de voz disponibles.

Tabla 4.17: Listado de comandos de Voz

<b>Comando</b>	<b>Función</b>
<i><b>DESPIERTA</b></i>	Inicia el proceso de reconocimiento. Llave del sistema.
<i><b>EN_LINEA</b></i>	Conecta el switch de corte.
<i><b>PRENDE</b></i>	Activa el pulsador de arranque.
<i><b>APAGA</b></i>	Desconecta el switch de corte.
<i><b>ASÍSTEME</b></i>	Brinda información sobre el uso del sistema.
<i><b>ESTADO</b></i>	Brinda información sobre el estado del sistema.
<i><b>BLOQUEATE</b></i>	Detiene el proceso de reconocimiento, desconecta el switch de corte.
<b>Elaborado por:</b> El Investigador.	

Todos los comandos de voz deben ser entonados de manera fuerte y clara a una distancia máxima de 5m de la motocicleta en un ambiente de hasta condiciones medias de ruido acústico ambiental.

Al ser un sistema de reconocimiento de voz dependiente de usuario, todos los comandos podrán ser ejecutados únicamente por el propietario del vehículo.

#### **4.11.4. COMANDOS DE RED**

El bloque de seguridad física cuenta con un módem 3G incorporado. La principal función de este dispositivo es el de permitir el bloqueo y localización remota de la motocicleta en caso de un posible robo. Los comandos viajan por la red celular, para garantizar compatibilidad con cualquier tipo de dispositivo móvil con conectividad 3G.

Existen dos tipos de comandos reconocidos por el sistema. El tipo “USER” y el tipo “ROOT”.

##### **4.11.4.1. COMANDO USER**

Son comandos que viajan a través de red celular y pueden ser enviados desde cualquier dispositivo con conectividad GSM. Manejan funciones que no implican factores de seguridad del sistema, por lo cual no prescinden de autenticación.

La tabla 4.18 resume los comandos “USER” y su función.

Tabla 4.18: Listado de comandos USER

<b>Comando</b>	<b>Función</b>
<i>STA</i>	Informa sobre el estatus actual del sistema.
<i>MOD</i>	Brinda información sobre algunos parámetros básicos de la motocicleta, como marca, modelo y placa.
<i>DEC</i>	Llama tres veces seguidas a tres números registrados. Para advertir de un posible siniestro sin incurrir en costos de red.
<i>BRD</i>	Envía mensajes de texto a los tres números registrados, solicitando ayuda en caso de siniestro o eventualidad.
<i>MAN</i>	Envía una versión condensada de los comandos User y su función.
<i>Sintaxis: nombre_comando &gt; enviar</i>	
<b>Elaborado por:</b> El Investigador.	

En caso de que ninguna coincidencia sea registrada por el sistema, se enviará un mensaje de error sugiriendo la utilización del comando MAN.

#### 4.11.4.2. COMANDOS ROOT

Este tipo de comandos están reservados para las operaciones de bloqueo y de localización remota. La estructura del comando maneja la estructura de User, pero un comando root debe ir acompañado de la contraseña de ascenso al sistema respectiva.

La tabla 4.19 resume los comandos “ROOT” y su función.

Tabla 4.19: Listado de comandos ROOT

<b>Comando</b>	<b>Función</b>
<i>POS</i>	Recupera la ubicación de la motocicleta.
<i>HLT</i>	Desactiva remotamente el switch de emergencia.
<i>Sintaxis: nombre_comando contraseña &gt; enviar</i>	
<b>Elaborado por:</b> El Investigador.	

Tanto los comandos de tipo USER como de tipo ROOT pueden ser enviados únicamente por un abonado, ya que los mensajes de texto recibidos son procesados mediante verificación de línea telefónica.

#### **4.11.4.3. FORMATO DE RESPUESTA MEDIANTE SMS**

Todos los comandos de tipo “user” y “root”, responden vía SMS al número que ha solicitado la ejecución de una orden específica. Los tiempos de respuesta dependen del tipo de comando ejecutado (POS y DEC son los que por su naturaleza registrarían los mayores tiempos en responder), sin embargo, el tiempo de respuesta ante un comando por parte del sistema no superará la barrera de los 120 sg.

El texto de confirmación enviado al remitente de la orden varía de comando a comando, siendo por lo general corto, pero manteniendo una línea común al final del mismo “...-- BAGHEERA”. Donde se confirma que la respuesta ha sido enviada desde el sistema electrónico de control de seguridad física.

Los caracteres “-- BAGHEERA”, hacen alusión al nombre clave del prototipo.

#### **4.11.5. DIAGRAMA DE FLUJO DEL BLOQUE DE CONTROL DE SEGURIDAD FÍSICA**

En la figura 4.13 se detalla el diagrama de flujo a seguirse para la implementación del código del sistema electrónico de control de seguridad física.



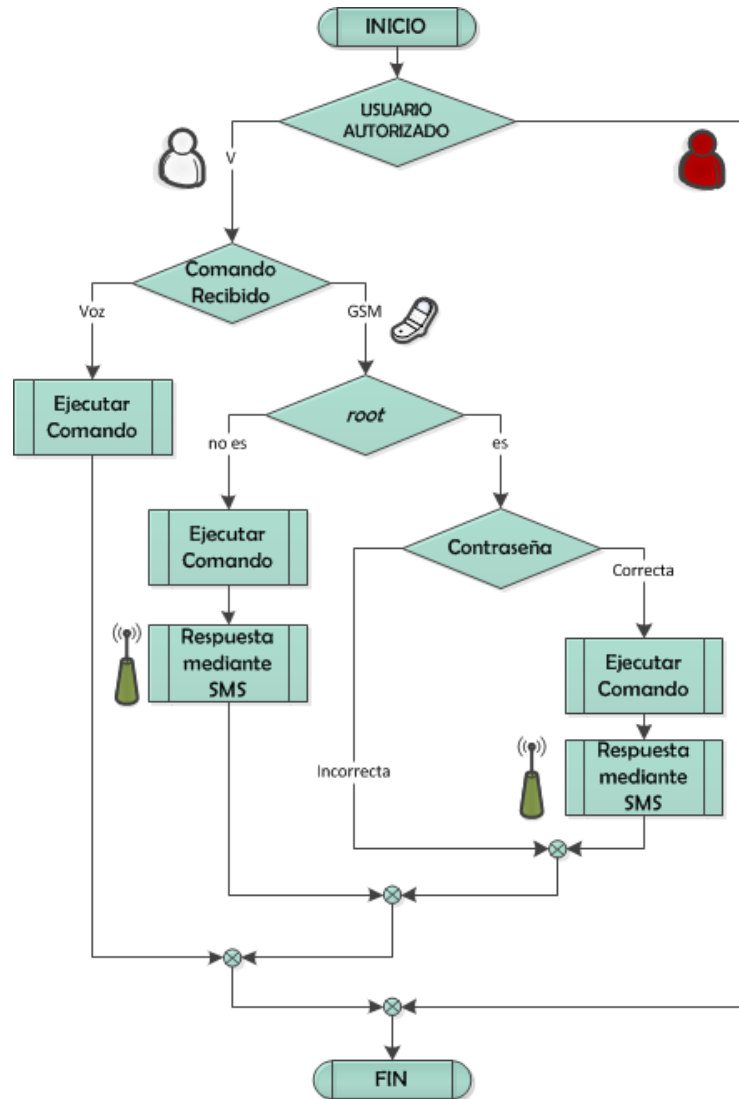


Figura 4.13: Diagrama de flujo para el bloque de seguridad  
**Elaborado por:** El Investigador.

El diagrama de la figura 4.13 detalla el funcionamiento del bloque de seguridad, el mismo que contempla dos escenarios:

- ▶ Identificación biométrica de usuario: únicamente el usuario cuya voz esté registrada en el sistema puede encender la motocicleta. Cualquier otra persona no puede acceder de ninguna manera.
- ▶ Recepción de comandos GSM: en caso de que el sistema reciba comandos mediante red celular, se procede a la verificación de línea y a una posterior verificación de contraseña para los comandos de tipo ROOT, previos al

procesamiento de la orden solicitada. Cualquier solicitud proveniente de una línea diferente a la registrada en el sistema es simplemente ignorada.

#### 4.12. BLOQUE DE CONDUCCIÓN ASISTIDA

El bloque de conducción asistida está enfocado al ahorro de combustible en la motocicleta, en base a la notificación de marchas a ser embragadas en función de umbrales de velocidad de bajas RPM (Revoluciones Por Minuto). Los valores obtenidos mediante las respectivas comprobaciones son listados en la tabla 4.20.

Tabla 4.20: Umbrales de velocidad para cambios de marcha de una motocicleta

MARCHA	UMBRAL
<i>1<sup>ra</sup></i>	<i>NA, arranque.</i>
<i>2<sup>da</sup></i>	<i>20 Km/h</i>
<i>3<sup>ra</sup></i>	<i>30 Km/h</i>
<i>4<sup>ta</sup></i>	<i>40 Km/h</i>
<i>5<sup>ta</sup></i>	<i>50 Km/h</i>
<b>Elaborado por:</b> El Investigador.	

Los valores listados en la prueba han sido comprobados mediante pruebas de manejo extensivas. Los mismo han sido determinados en una zona óptima de revoluciones del motor (4000 RPM's), que brindan el torque necesario para entrar a la marcha siguiente sin problema alguno, manteniendo constante la aceleración [80].

A su vez, mantener un bajo régimen de revoluciones el motor disminuye de manera considerable el consumo de combustible así como la emisión de gases contaminantes [81].

##### 4.12.1. DISEÑO DEL BLOQUE DE CONDUCCIÓN ASISTIDA

El bloque de conducción asistida interactuará con la señal del velocímetro. Se pretende informar al conductor mediante el uso de las interfaces de salida del momento adecuado

para subir o bajar a la marcha adecuada en base a la presencia o ausencia de un valor de umbral de velocidad.

- ▶ Señal a analizar: digital pulsante de baja frecuencia.

En la figura 4.14, se puede apreciar el diagrama eléctrico del circuito velocímetro – tacómetro de una motocicleta.

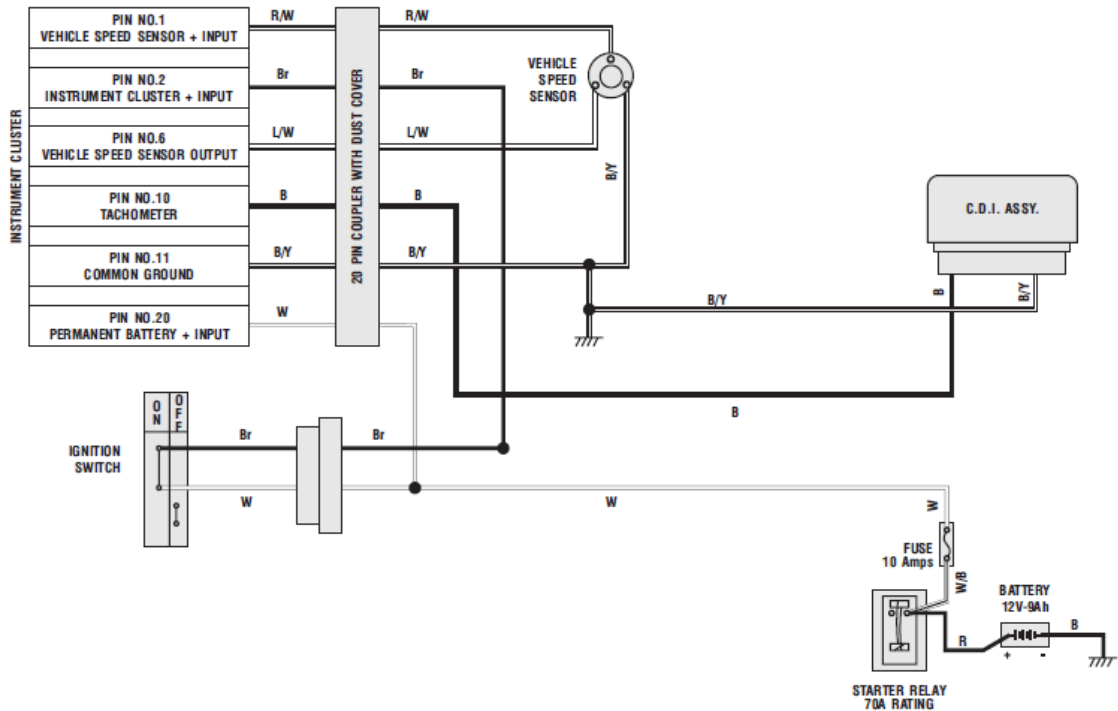


Figura 4.14: Conjunto velocímetro – tacómetro de una motocicleta

**Fuente:** Pulsar DTSi Reparation Manual [82].

Como se puede apreciar, el sensor es de tipo digital y consta básicamente de tres terminales, alimentación, tierra y salida; los cuales se encuentran conectados a la unidad electrónica de control.

El sensor de velocidad utilizado es de efecto Hall. La operación del sensor se basa en la detección de pulsos generados mediante la presencia de campos magnéticos. La señal es entregada al velocímetro mediante un solo cable, el cual muestra la velocidad en KM/H. En la figura 4.15, se puede apreciar el aspecto físico del sensor utilizado.


	<b>Part Name</b>	Wheel Sensor Unit
	<b>Part No.</b>	DK 1011 63
	<b>Description</b>	New part added. It is a sensor sensing the vehicle speed and is a non contact type wheel sensor.
	<b>Identification Mark</b>	NIL

Figura 4.15: Sensor de velocidad de Efecto Hall de una motocicleta

**Fuente:** Pulsar DTSi Reparation Manual [83].

Realizadas las pruebas respectivas, se opta por la conexión directa de la salida del sensor a una GPIO de la placa Arduino Leonardo. Este criterio de diseño se fundamenta en dos detalles de la señal entregada por el sensor:

- ▶ Es de tipo pulsante a nivel digital, ideal para ser analizada directamente con una función de conteo.
- ▶ Se ha verificado que se la señal de salida del sensor puede alimentar al cuadro de instrumentos y la GPIO de Arduino a la vez. Esta característica que simplifica el diseño del sistema de conducción asistida se debe a la naturaleza de la señal, al ser digital es perfectamente reconocible y tratable a pesar de ser atenuada, siempre y cuando la misma se encuentre dentro de los umbrales de reconocimiento de sus niveles lógicos (al menos 2.7V para un uno lógico).

En la figura 4.16 se muestra el circuito de pruebas utilizado para el desarrollo del sistema de conducción asistida.

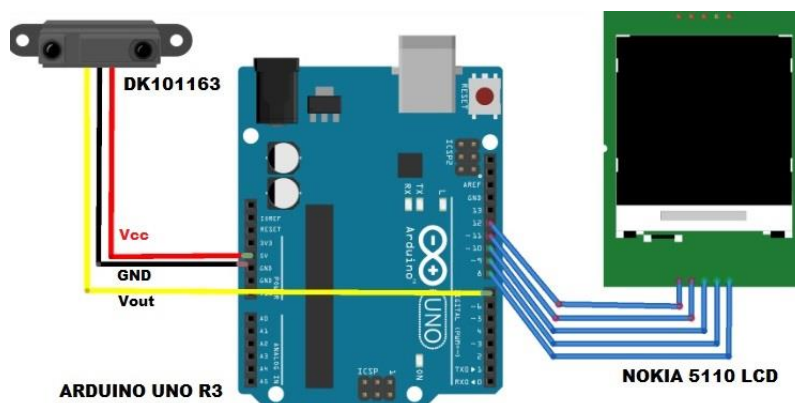


Figura 4.16: Circuito de pruebas bloque de conducción asistida

**Elaborado por:** El Investigador.

#### 4.12.2. DIAGRAMA DE FLUJO DEL BLOQUE DE CONDUCCIÓN ASISTIDA

En la figura 4.17 se detalla el diagrama de flujo a seguirse para la implementación del código del sistema electrónico de conducción asistida.

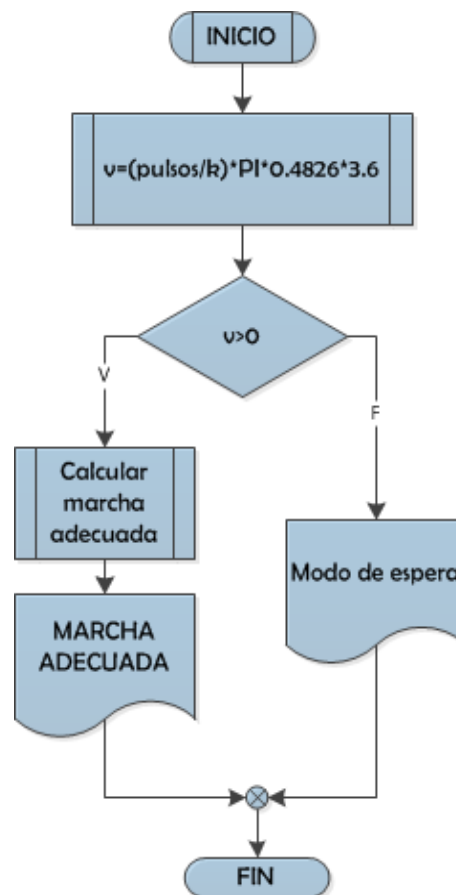


Figura 4.17: Diagrama de flujo para el sistema de conducción asistida

**Elaborado por:** El Investigador.

El análisis de la figura 4.17, detalla el comportamiento del bloque de conducción asistida, mismo que entra en operación cuando la motocicleta está en movimiento. El sistema procede a calcular la velocidad de circulación del vehículo en unidades métricas y en caso de registrarse un valor de umbral, notifica al usuario una marcha adecuada del motor para la velocidad actual.

## 4.13. IMPLEMENTACIÓN DEL PROTOTIPO

### 4.13.1. ALIMENTACIÓN DEL SISTEMA

Como primer paso, se procede a determinar los consumos de corriente y tensiones nominales de operación de las placas que constituyen el prototipo del sistema para la selección una fuente adecuada, tal y como se aprecia en la tabla 4.21.

Tabla 4.21: Tensiones y corrientes de operación de los elementos del sistema

ELEMENTO	VOTAJE DE ALIMENTACIÓN	CONSUMO DE CORRIENTE
Arduino Mega	7 – 12 V	70mA
Arduino Uno	7 – 12 V	45mA
SIMCOM 908 KITS UNO	3.2 – 4.8 V	Hasta 1A en arranque 75mA en acceso inicializado
NOKIA LCD 5110	5V	7mA
EASY VR 2.0	4.8 – 5.5 V	30mA
Circuito de Actuación	5V	150mA (estimado)
<b>Elaborado por:</b> El Investigador.		

Los valores de corriente detallados en la tabla son los máximos consumos provistos por la documentación facilitada por cada uno de sus respectivos fabricantes.

Las tensiones de operación de los módulos EasyVR, SIMCOM 9078 y Nokia LCD 5110 son entregadas por el circuito de regulación interno de los módulos Arduino.

Se procede a realizar los cálculos de disipación de potencia utilizando un LM7809, para trabajar con una fuente de alimentación de 9V. No es posible utilizar un LM7812, debido a que la tensión nominal de la batería incluso con el motor encendido (14.5 V), no está 3V por encima del recomendado para que el circuito funcione adecuadamente. El cálculo de la potencia disipada en forma de calor por los reguladores de voltaje viene dado por la relación descrita en la ecuación 4.1:

$$P = (V_{in} - V_{out}) * I \quad (4.1)$$

Reemplazando por los valores de la tabla 4.21, para la operación inicializada del módulo SIMCOM 908 se tiene que:

$$P_{REG}[W] = (14.5 - 9)[V] * (70 + 45 + 75 + 30 + 7 + 150)[mA]$$

$$P_{REG}[W] = 2.0735W$$

Reemplazando por los valores de la tabla, para la operación de arranque del módulo SIMCOM 908 se tiene que:

$$P_{REG}[W] = (14.5 - 9)[V] * (0.07 + 0.045 + 1 + 0.03 + 0.007 + 0.15)[A]$$

$$P_{REG}[W] = 7.161W$$

Se determina que no es factible el diseño de una fuente mediante la utilización de la batería de la motocicleta por dos razones:

- ▶ La corriente requerida por el sistema supera 1A para el arranque en frío del módulo SIMCOM 908.
- ▶ La excesiva potencia disipada.

Se opta entonces por utilizar una batería recargable de 12V y 3AH. De esta forma, el diseño de la fuente presenta estas nuevas características:

- ▶ Las placas Arduino pueden ir conectadas directamente a la batería, por lo cual, son excluidos del cálculo de la potencia disipada. Una tensión nominal de 12V se encuentra dentro del rango recomendado de alimentación. El diseño de la caja provee la ventilación adecuada para evitar el sobrecalentamiento de las placas.
- ▶ A su vez, la regulación de voltaje y dotación de corriente para los shields (EASY VR 2.0 y SIMCOM 908) así como para el display son manejadas directamente por los reguladores propios de las placas.

Resta de esta forma calcular la corriente disipada por el circuito de actuación. Como el mismo trabaja a 5V, se utilizará un LM7805 a continuación de un LM7809. Se procede a calcular la potencia disipada por cada uno de los elementos.

$$P_{7809}[W] = (12 - 9)[V] * 0.15[A]$$

$$P_{7809}[W] = 0.45W$$

$$P_{7805}[W] = (9 - 5)[V] * 0.15[A]$$

$$P_{7805}[W] = 0.6W$$

La potencia consumida, no requiere de métodos de disipación, y la corriente requerida se encuentra dentro del rango nominal de los reguladores.

#### 4.13.2. PLACA DE ACTUADORES

Es necesaria la construcción de una placa de actuadores para que los controles de la mano derecha de la motocicleta de prueba puedan ser controlados por las salidas de los sistemas microcontrolados.

##### 4.13.2.1. SIMULACIONES

La figura 4.18 detalla el circuito de simulación utilizado para la elaboración de la placa de actuadores.

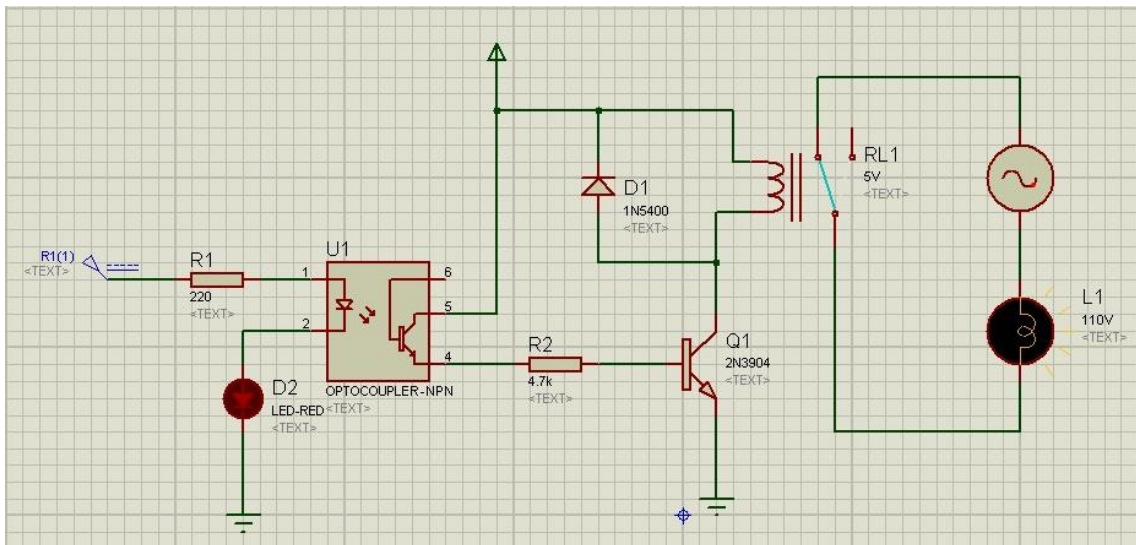


Figura 4.18: Simulación del circuito de la placa de actuadores  
**Elaborado por:** El Investigador.

La figura 4.18 permite apreciar que se ha optado por la selección de una interfaz de actuación mediante relés con una etapa opto aislada y otra transistorizada. De esta manera, se logra digitalizar los controles del switch de corte y del pulsador de arranque. Además, se ha colocado un diodo 1N5408 (corriente nominal 3A) en anti paralelo a la



bobina del relé para evitar el retorno de cualquier posible transitorio generado en la bobina al momento de realizarse el enclavamiento de la misma.

Los anexos B6 y B7 ofrecen las hojas de datos del opto acoplador 4N35 y del diodo 1N5408 respectivamente.

#### 4.13.2.2. DISEÑO DE LA PLACA DE ACTUADORES

Se procedió a realizar el diseño del circuito de actuación para la interacción del sistema tanto con el switch de emergencia como con el pulsador de ignición. Los diagramas tanto esquemáticos como pictóricos son ilustrados en la figura 4.19 y en la figura 4.20, respectivamente.

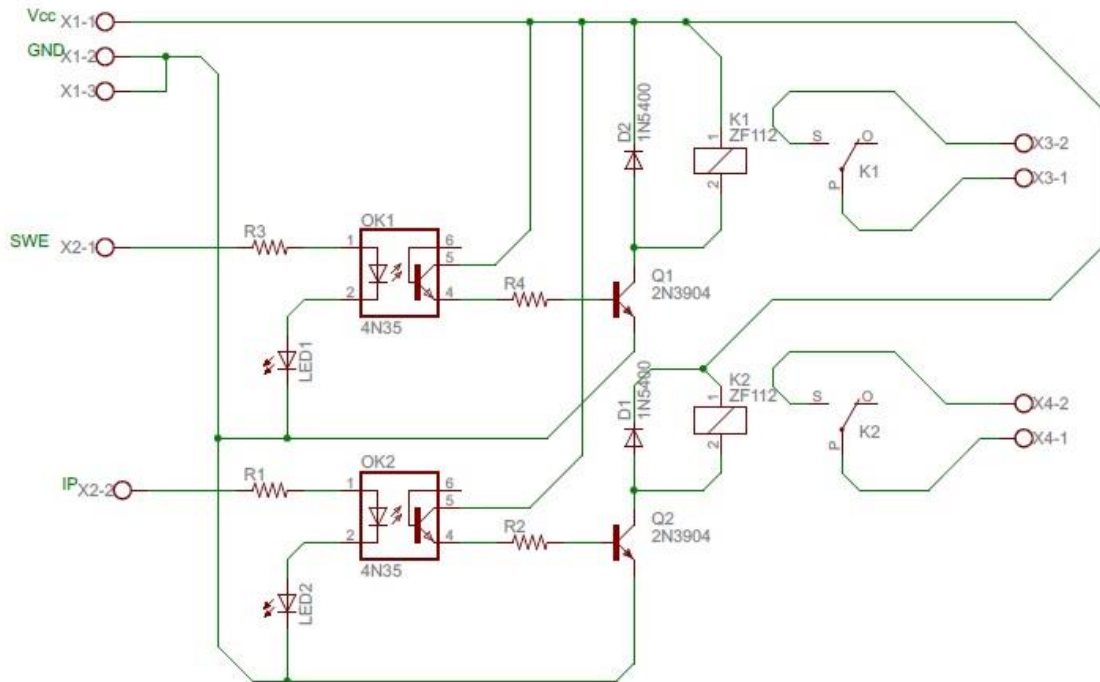


Figura 4.19: Diagrama esquemático circuito de actuación  
**Elaborado por:** El Investigador.

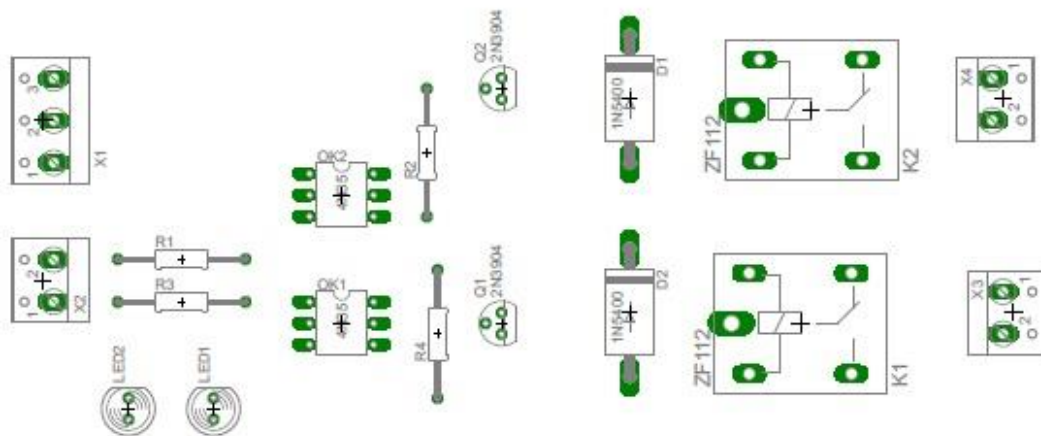


Figura 4.20: Diagrama pictórico circuito de actuación  
**Elaborado por:** El Investigador.

La inspección del diagrama esquemático mostrado en la figura 4.19, muestra que se optó por un diseño basado en la utilización de opto acopladores con una subsecuente etapa transistorizada. Los diodos en anti paralelo protegen al circuito de cualquier corriente reversa producida por el enclavamiento de los relés. Las figuras 4.21 y 4.22 muestran las pistas impresas y la placa de actuadores ya ensamblada respectivamente.

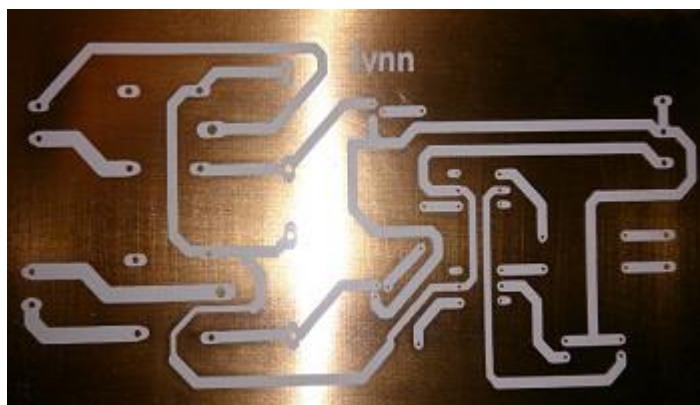


Figura 4.21: Pistas adheridas a la baquela  
**Fuente:** El Investigador.

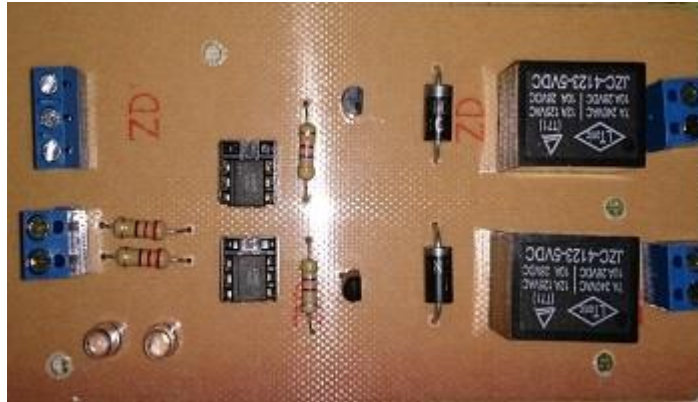


Figura 4.22: Placa de actuadores ensamblada  
Fuente: El Investigador.

El anexo B8 provee el diagrama esquemático y el ruteado de las pistas para la placa de actuadores.

#### 4.13.3. CIRCUITO FINAL

La figura 4.23 muestra el diagrama de bloques del prototipo final en base a las consideraciones de las secciones del bloque de seguridad y del bloque de conducción asistida.

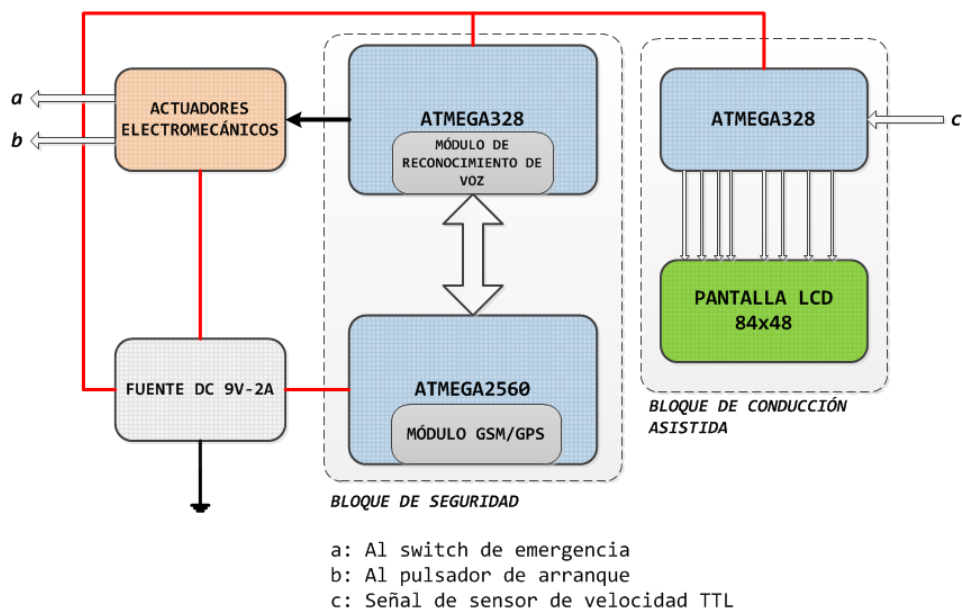


Figura 4.23: Diagrama de bloques del prototipo  
Fuente: El Investigador.

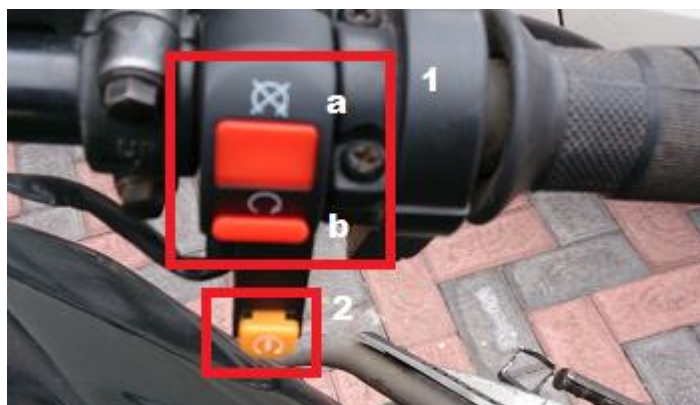
Del análisis de la figura 4.23, se aprecia que el sistema está conformado por dos bloques principales:

- ▶ Bloque de seguridad: conformado por dos sistemas microcontroladores. El sistema de reconocimiento es manejado por el ATMEGA328 (Arduino Uno SMD), mientras que el sistema GSM/GPS es manejado por el ATMEGA2560 (Arduino Mega). Este bloque es el encargado de la digitalización de los controles de encendido, el manejo de comando de voz y de red, así como del apagado remoto de la motocicleta mediante SMS.
- ▶ Bloque de conducción asistida: conformado por un sistema microcontrolador ATMEGA328 (Arduino Uno SMD) y un módulo LCD de 84x48. Este bloque es el encargado de indicar marchas adecuadas para reducir el consumo de combustible, en base a la lectura de la velocidad de circulación de la motocicleta.

Ambos bloques poseen un funcionamiento independiente el uno del otro. La activación del bloque de conducción asistida y del sistema GSM/GPS del bloque de seguridad son opcionales. La activación del sistema de reconocimiento de voz es obligatoria, ya que este maneja el encendido eléctrico de la motocicleta.

#### **4.13.4. ADECUACIONES DEL CABLEADO DE LA MOTOCICLETA**

La figura 4.24 detalla los controles de la mano derecha de la motocicleta seleccionada para las pruebas del sistema. Los controles de la mano derecha son los actuadores mecánicos del encendido eléctrico.



- 1: Switch de emergencia
  - a-) Posición de apagado
  - b-) Posición de arranque
- 2: Pulsador de ignición

Figura 4.24: Controles de la mano derecha de una motocicleta  
**Fuente:** El Investigador.

Las figuras 4.25 - 4.27 muestran en detalle las inspecciones realizadas en torno a la batería de la motocicleta.



Figura 4.25: Ubicación de la batería en la motocicleta de prueba  
**Fuente:** El Investigador.



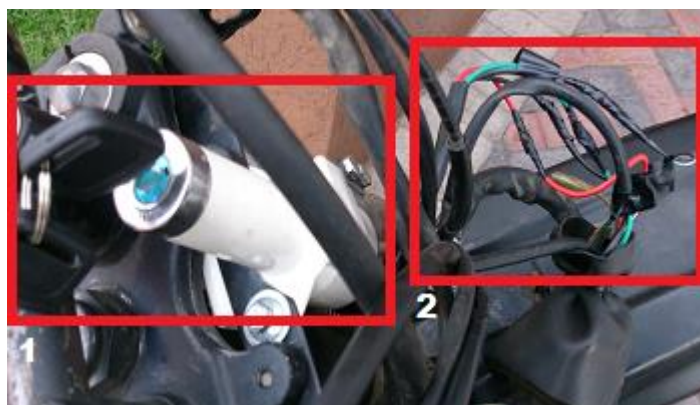
Figura 4.26: Tensión nominal de la batería con el motor apagado  
**Fuente:** El Investigador.



Figura 4.27: Tensión nominal de la batería con el motor encendido  
**Fuente:** El Investigador.

En la figura 4.26 se aprecia una tensión de 12.69V presente en la batería cuando la motocicleta se encuentra apagada. El rango de valores aceptables se sitúa entre [12,13] V, variando ligeramente de batería a batería. La figura 4.27 muestra una tensión de 14.58V cuando la motocicleta se encuentra encendida. La tensión se eleva aproximadamente 2V debido a que la batería actúa como carga de la unidad rectificadora – reguladora. Si la tensión de la batería es de [12,13] V con la motocicleta encendida, esto es indicio de una falla en la batería o en última instancia de un problema con la unidad de carga.

Las figuras 4.28 – 4.31 muestran el detalle del cableado principal de la motocicleta, así como el despiece de los controles de la mano derecha.



- 1: Columna de ignición
- 2: Cableado principal

Figura 4.28: Identificación del cableado de la columna de ignición  
**Fuente:** El Investigador.



- a: Cables del switch de emergencia, par verde – blanco/negro  
 verde: masa, cable de bajada  
 blanco/negro: cable de subida
- b: Cables del pulsador de ignición, par verde – rojo /amarillo  
 verde: masa, cable de bajada  
 rojo/amarillo: cable de subida

Figura 4.29: Identificación de los pares del switch de emergencia  
 y del pulsador de arranque  
**Fuente:** El Investigador.



Figura 4.30: Desmontaje del interruptor de la mano derecha  
**Fuente:** El Investigador.

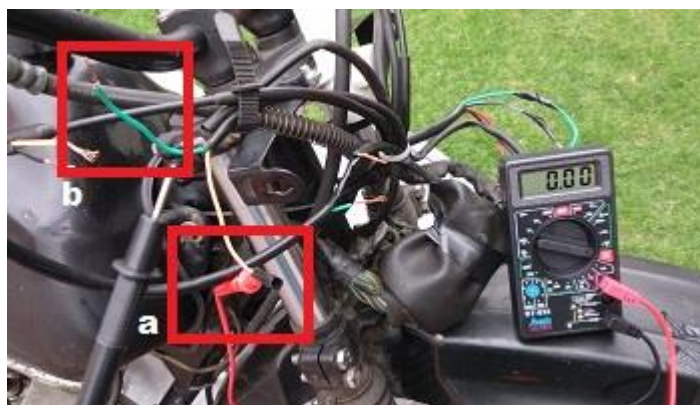


a: blanco/negro: cable de subida  
b: verde: masa, cable de bajada

Figura 4.31: Cortocircuito del switch de emergencia  
**Fuente:** El Investigador.

Las figuras 4.32 – 4.33 muestran los procesos asociados a la medición de la corriente de ignición que circula por los conductores del pulsador de arranque. La determinación de esta corriente de 1.5A ha sido fundamental para el desarrollo de la placa de actuadores. La figura 4.34 detalla las primeras pruebas de encendido realizadas con una placa de relés.





a: rojo/amarillo: cable de subida  
 b: verde: masa, cable de bajada

Figura 4.32: Desmontaje de los cables del pulsador de arranque  
**Fuente:** El Investigador.



$$I \cong 1.5 [A]$$

Figura 4.33: Corriente de ignición de encendido  
**Fuente:** El Investigador.



Figura 4.34: Pruebas de ignición mediante relés  
**Fuente:** El Investigador.

#### 4.13.5. MONTAJE DE LA CAJA E INSTALACIÓN

Las figuras 4.35 – 4.37, muestran algunos de los pasos relacionados al proceso de elaboración de la caja para la ubicación de las placas.



Figura 4.35: Ensamble de la caja con perforaciones para cableado y ventilación  
**Fuente:** El Investigador.

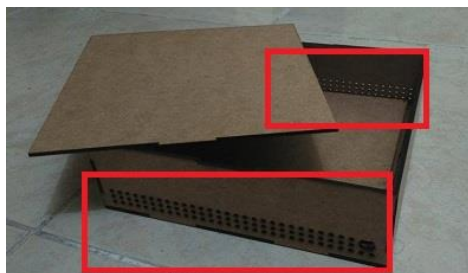


Figura 4.36: Disposición de las ranuras de ventilación de la caja  
**Fuente:** El Investigador.



Figura 4.37: Caja del prototipo  
**Fuente:** El Investigador.

La figura 4.38 muestra el aspecto final del prototipo con el circuito instalado en la caja.



Figura 4.38: Prototipo instalado en la caja  
**Fuente:** El Investigador.

La figura 4.39 muestra el prototipo montado en el modelo de pruebas.



Figura 4.39: Montaje del prototipo en la motocicleta  
**Fuente:** El Investigador.

#### **4.14. PRUEBAS DEL PROTOTIPO**

Se procede en la presente sección a analizar los resultados de las pruebas de funcionamiento del prototipo de sistema electrónico de control de seguridad física y conducción asistida de una motocicleta.

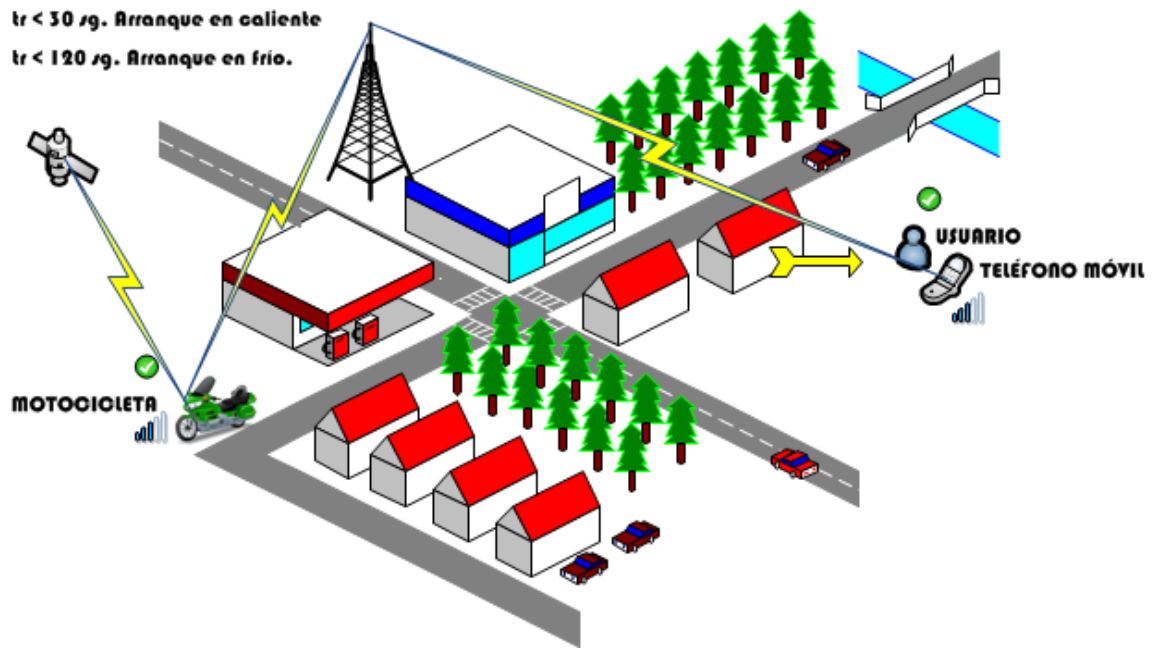
#### **4.14.1. PRUEBAS DEL SISTEMA ELECTRÓNICO DE CONTROL DE SEGURIDAD FÍSICA**

Las pruebas realizadas señalan que la conexión del switch de emergencia mediante un control biométrico por voz es altamente satisfactoria.

Se ha determinado que las capacidades de reconocimiento de comandos por parte del sistema biométrico se ajustan perfectamente a las condiciones normales de ruido ambiental, siendo posible incluso dar órdenes desde una distancia de s hasta 4m del micrófono, utilizando un tono de voz firme y sonoro.

Las pruebas realizadas en lo que concierne al sistema de rastreo y bloqueo remoto, arrojan también resultados positivos. La conexión y desconexión del switch de emergencia mediante los comandos tipo root: POW y HLT, es oportuna y de acción inmediata. Se considera que un bloqueo remoto ejecutado desde un terminal verificado en caso de robo con el motor en marcha tendría un 100% de efectividad, haciendo que sea imposible arrancar el motor para nadie que no sea el propietario del vehículo.

El bloque de rastreo cumple las expectativas de diseño y promete acorde a sus especificaciones, tomando alrededor de 90 segundos en ser atendida una solicitud de rastreo de posición mediante el comando POS con el sistema arrancando en frío (es decir, la primera solicitud de rastreo registrada desde que inicio todo el sistema electrónico de control). Las solicitudes consecuentes de posición (arranque en caliente), sólo demoran el tiempo en que le toma al comando viajar por la red celular disponible y recibir una respuesta del sistema electrónico de control (aproximadamente 30 segundos), tal y como se ilustra en la figura 4.40.



$t_r$ : tiempo de respuesta.

Figura 4.40: Tiempos de respuesta estimados del prototipo  
**Elaborado por:** El Investigador.

No existe un límite de distancia entre el usuario y la motocicleta, siempre que ambos que encuentren en una zona con cobertura GSM.

#### 4.14.2. PRUEBAS DEL SISTEMA ELECTRÓNICO DE CONDUCCIÓN ASISTIDA

Mediante la ejecución de las pruebas pertinentes, se ha determinado que el funcionamiento del bloque de conducción asistida es satisfactorio.

Los resultados de la tabla 4.22 están estipulados en pruebas realizadas en un recorrido diario de 10.8 KM, mismo que contempla tramos rectos, pendientes y descensos, avalando la compatibilidad práctica de los resultados con la conducción en la vida real

Tabla 4.22: Comparativa de consumo de combustible en una motocicleta sin y con sistema electrónico de conducción asistida		
Día	Consumo (Galones)	
	Sin sistema de conducción asistida	Con sistema de conducción asistida
1	0.2	0.14
2	0.19	0.12
3	0.21	0.14
4	0.21	0.14
5	0.22	0.13
6	0.19	0.12
7	0.2	0.14
8	0.2	0.13
9	0.21	0.12
10	0.19	0.13
11	0.2	0.13
12	0.21	0.14
13	0.21	0.12
14	0.19	0.13
≅TOTAL	2.88	1.92
<b>Elaborado por: El Investigador.</b>		

Los datos de la tabla 4.22 pueden ser apreciados de manera gráfica en la figura 4.41.

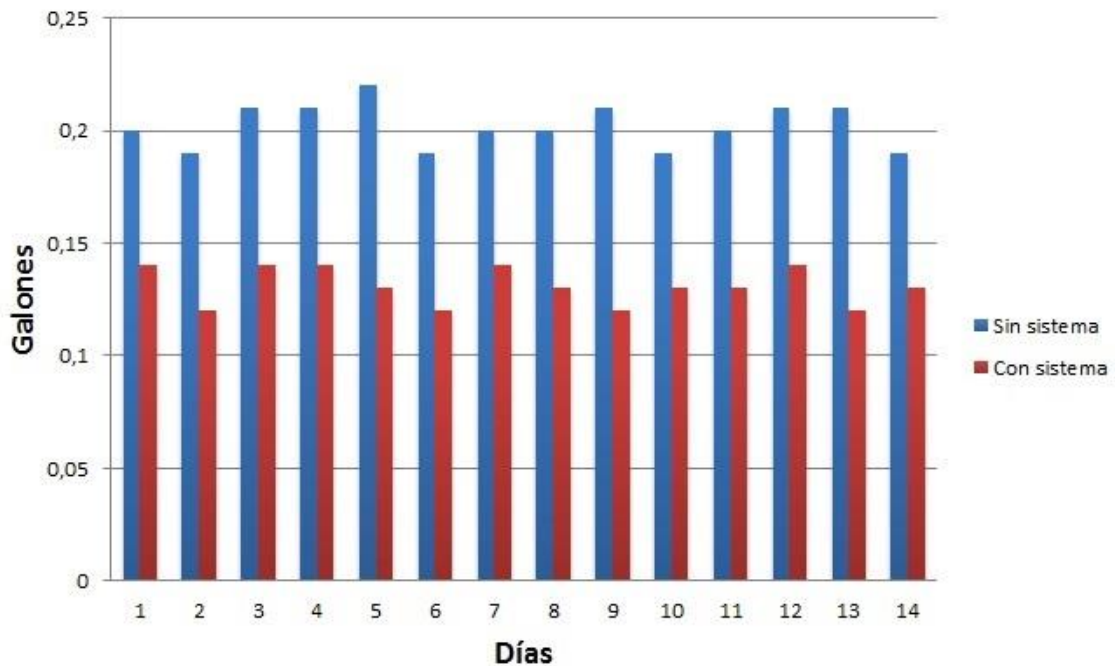


Figura 4.41: Comparativa de consumo de combustible en una motocicleta sin y con sistema electrónico de conducción asistida  
**Elaborado por:** El Investigador.

Del análisis de la tabla 4.22 y de la figura 4.41, se confirma un notable decremento en el consumo de combustible, esto debido al algoritmo del sistema de conducción asistida, mismo que sugiere cambios en zonas bajas de revoluciones del motor.

El costo del galón de gasolina súper se encuentra en los USD 2.084. Se procede a calcular el coste anual del combustible en una motocicleta sin sistema de conducción asistida.

$$1.44 \text{ gal} * 2.084 \frac{\text{USD}}{\text{gal}} \cong \text{USD } 3.00 \text{ semanales} \cong \text{USD } 156.00 \text{ anuales}$$

Se procede a calcular el coste anual del combustible en una motocicleta con sistema de conducción asistida.

$$0.96 \text{ gal} * 2.084 \frac{\text{USD}}{\text{gal}} \cong \text{USD } 2.00 \text{ semanales} \cong \text{USD } 104 \text{ anuales}$$

Los resultados de las pruebas del sistema de conducción asistida denotan un ahorro de aproximadamente USD 52.00 anuales para un recorrido diario de aproximadamente 10.8 Km. Se determina entonces que una motocicleta que utilice el sistema consumirá el 66% del combustible que consuma una motocicleta que no lo posea.

Se determina entonces que el funcionamiento del sistema electrónico de conducción asistida es satisfactorio y permite al usuario ahorrar en costes en cuanto a combustible se refiere.

#### 4.15. PRESUPUESTO DEL PROYECTO

La tabla 4.23 muestra el detalle del presupuesto utilizado para la implementación del prototipo del sistema.

Tabla 4.23: Presupuesto del proyecto

ELEMENTO	UNIDAD	CANTIDAD	COSTE UNITARIO (USD)	TOTAL (USD)
Arduino Mega 2560 R3	c/u	1	38.00	38.00
Arduino Uno SMD R3	c/u	2	24.00	48.00
EasyVR Shield 2.0	c/u	1	65.00	65.00
SIM900 KITS UNO	c/u	1	120.00	120.00
Nokia LCD Display	c/u	1	9.99	9.99
Dakar User Manual (Copia)	hoja	42	0.02	0.84
Pulsar DTSi Reparation Manual (Copia)	hoja	60	0.02	1.20
½ Plancha MDF de 3 líneas	c/u	1	6.00	6.00
Corte láser	c/u	1	10.00	10.00
Trabajo de pintura	c/u	1	8.00	8.00
Bisagras	par	2	0.50	1.00
Tornillos de 3mm	c/u	16	0.01	0.16
Tuercas de 3mm	c/u	16	0.01	0.16
Impresión plotter de corte para logotipos	c/u	3	2.50	7.50



Impresión plotter de corte para pistas	c/u	1	5.00	5.00
Baquela 10X20	c/u	1	1.40	1.40
Relé 5V	c/u	2	0.50	1.00
1N5408 (Diodo de 3A)	c/u	2	0.10	0.20
4N35 (Opto acoplador)	c/u	2	0.60	1.20
Borneras de 2 pines	c/u	5	0.25	1.25
Batería 12V 3AH	c/u	1	12.00	12.00
Lagartos	par	4	0.30	1.20
Jack de poder	c/u	4	0.25	1.00
Cable flexible #18	metros	3	0.30	0.90
Cable flexible #16	metros	3	0.45	1.35
Reed Switch (Contacto magnético simple)	c/u	1	5.60	5.60
Resma de papel A4	c/u	1	4.00	4.00
Diseño preliminar del prototipo	horas	6	2.38	14.28
Diseño final del prototipo	horas	10	2.38	23.80
<b>TOTAL (USD)</b>				<b>390.03</b>
<b>Elaborado por: El investigador.</b>				

El valor del diseño preliminar y final del prototipo han sido calculados en base a la información proporcionada por la tabla de sueldos mínimos sectoriales 2015 suministrada por el ministerio de relaciones laborales.

## **CAPÍTULO V**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **5.1 CONCLUSIONES**

- ▶ La aplicación de tecnologías de plataformas de desarrollo de hardware libre basadas en microcontroladores AVR permiten digitalizar los controles del sistema CDI de una motocicleta.
- ▶ Se ha determinado que la utilización conjunta de módulos de identificación biométricos con módulos de conectividad GSM/GPS; incrementan exponencialmente la seguridad física de una motocicleta frente a posibles escenas de hurtos.
- ▶ Se concluye que la utilización de algoritmos adecuados para el monitoreo de la velocidad de una motocicleta, permiten reducir de manera considerable el consumo de combustible, la emisión de gases contaminantes; y alargar el tiempo de vida útil del sistema de lubricación interna del motor.

#### **5.2 RECOMENDACIONES**

- ▶ Se recomienda la utilización de plataformas de desarrollo basadas en AVR para la implementación de sistemas de control o cualquier sistema embebido. Esto debido a su elevada flexibilidad, compatibilidad, poder de procesamiento y funcionalidades añadidas frente a las limitaciones natas de la utilización de un simple microcontrolador de la familia PIC.

- ▶ Se recomienda la inclusión de sistemas de control y rastreo para todo propietario de una motocicleta de ensamblaje nacional. Las limitadas capacidades de las unidades electrónicas de control incluidas se pueden ver potenciadas de manera importante mediante módulos compatibles.
- ▶ Se recomienda estudiar a fondo las características de los módulos fabricados por terceros. Arduino es una plataforma de desarrollo en constante evolución, hecho que colateralmente afecta de manera temporal o permanente las características de compatibilidad con algunos módulos debido a bugs en las librerías asociadas o discrepancias en los niveles lógicos manejados por los protocolos de comunicación.

## Bibliografía

[1] Heil Carsten, “1000 Motos”, 1<sup>ra</sup> ed, Editorial NGV, 2013.

[2] Cando Luis, 2011. “Bloqueo electrónico en el encendido de un vehículo, para brindar un sistema de seguridad contra robos”. Repositorio Digital UTA. Tesis de grado.

**Disponible en:**

<http://repo.uta.edu.ec/handle/123456789/444>

[3] Tello Pantoja, Doris Alexandra, 2007. “Diseño y construcción de un módulo automático de automóvil ortopédico para uso de una persona con discapacidad para desplazarse”. Repositorio EPN. Tesis de grado.

**Disponible en:**

<http://bibdigital.epn.edu.ec/handle/15000/9884>

[4] Allan W. M. Bonnick, “Automotive Computer Controlled Systems”. 1<sup>ra</sup> ed, Butterwoth Heinneman, 2001.

**Disponible en:**

[http://setcom.ee/tanno/info/is/teave/ained/ele/ele\\_car\\_butterworth\\_bonnick\\_automotive\\_computer\\_controlled\\_systems\\_2001.pdf](http://setcom.ee/tanno/info/is/teave/ained/ele/ele_car_butterworth_bonnick_automotive_computer_controlled_systems_2001.pdf)

[5] Wiring Diagrams, “Aprilia RSV Mille Ignition System Wiring Diagram”. Extraído en 04-15, desde:

<http://www.wiringdiagrams21.com/2009/04/30/aprilia-rsv-mille-ignition-system-wiring-diagram/>

[6] Ccs Labs, “Electronic Control Units”. Extraído en 04-15, desde:

<http://www.ccs-labs.org/teaching/c2x/2014s/05-ecus.pdf>

[7] BMW Motorrad, “R1200GS Adventure Rider’s Manual (US)”. 2014.

**Disponible en:**

[http://www.bmwmotorcycles.com/us/en/index.html?content=http://www.bmwmotorcycles.com/us/en/services/manuals/manuals\\_main.html&notrack=1](http://www.bmwmotorcycles.com/us/en/index.html?content=http://www.bmwmotorcycles.com/us/en/services/manuals/manuals_main.html&notrack=1)

[8] Iván Cisneros Rodríguez, Cisneros, “Tu taller Mecánico”. Revista mecánica. Fascículo 5, 2012.

[9] UNNOMOTORS, “*Dakar User’s Manual*”. Manual de Usuario. 1<sup>ra</sup> ed, 2013.

[10] BMW Education, “*Automotive Sensors*”. Workbook.

Disponible en:

[http://www.bmweducation.co.uk/academy/downloads/L1\\_Auto\\_Sensors\\_Accessible.pdf](http://www.bmweducation.co.uk/academy/downloads/L1_Auto_Sensors_Accessible.pdf)

[11, 12] Aficionados a la mecánica. “*Sensores en el Automóvil*”. Extraído en 12-14 desde:

<http://www.aficionadosalamecanica.net/sensores.htm>

[13] BOSCH, 2002. “*Los sensores en el automóvil*”. Manual de Instrucción Técnica. Bosch, 2002.

[14] William J. Fleming, “Overview of Automotive Sensors”. IEEE Sensors Journal. Volumen 1, N<sup>o</sup> 4. 2001.

[15, 16] Maloney Timothy, “*Electrónica Industrial Moderna*”. 5<sup>ed</sup>, Person Prentice Hall, 2006.

[17] William J. Fleming, “Overview of Automotive Sensors”. IEEE Sensors Journal. Volumen 1, N<sup>o</sup> 4. 2001.

[18] BMW Education, “*Automotive Sensors*”. Workbook.

Disponible en:

[http://www.bmweducation.co.uk/academy/downloads/L1\\_Auto\\_Sensors\\_Accessible.pdf](http://www.bmweducation.co.uk/academy/downloads/L1_Auto_Sensors_Accessible.pdf)

[19] Pérez Diego. “*Sistemas Electrónicos*”. Sensor de Efecto Hall, 2012. Extraído en 11-14 desde:

<http://se2amm.blogspot.com/2012/05/sensor-de-efecto-hall.html>

[20] Eric Rubow, “*Open Source Hardware*”, 2008.

**Disponible en:**

[http://cseweb.ucsd.edu/classes/fa08/cse237a/topicresearch/erubow\\_tr\\_report.pdf](http://cseweb.ucsd.edu/classes/fa08/cse237a/topicresearch/erubow_tr_report.pdf)

[21] The Uzebox Project. Uzebox Showcase. Extraído en 12-14 desde:

<http://belogic.com/uzebox/index.asp>

[22] Alicia Gibb, “*Building Open Source Hardware*”. 1<sup>ra</sup> ed, Editorial Addison Wesley, 2014.

[23] Ucamco, “*The Gerber Format Specification*”. Revisión 06-2015.

**Disponible en:**

[http://www.ucamco.com/files/downloads/file/81/the\\_gerber\\_file\\_format\\_specification.pdf](http://www.ucamco.com/files/downloads/file/81/the_gerber_file_format_specification.pdf)

[24] Roberto Acosta, “*Open Source Hardware*”. Tesis para la obtención de Máster en Ciencias de Ingeniería y Control del Instituto de Tecnología de Massachusetts, 2009.

[25] Alessandro D’ Ausilio, “*Arduino: A low-cost multipurpose lab equipment*”.

**Disponible en:**

<http://www.ncbi.nlm.nih.gov/pubmed/22037977>

[26] Atmel, “*ATMEGA328P Datasheet*”.

**Disponible en:**

<http://www.atmel.com/images/doc8161.pdf>

[27] Atmel, “*ATMEGA32u4 Datasheet*”.

**Disponible en:**

<http://www.atmel.com/images/doc7766.pdf>

[28] Atmel, “*ATMEGA2560 Datasheet*”.

**Disponible en:**

<http://www.atmel.com/images/doc2549.pdf>

[29] Atmel, “*SAM3X/SAM31 Series Datasheet*”.

**Disponible en:**

[http://www.atmel.com/Images/Atmel-11057-32-bit-Cortex-M3-Microcontroller-SAM3X-SAM3A\\_Datasheet.pdf](http://www.atmel.com/Images/Atmel-11057-32-bit-Cortex-M3-Microcontroller-SAM3X-SAM3A_Datasheet.pdf)

[30] Intel, “*Galileo Datasheet*”.

**Disponible en:**

[http://download.intel.com/support/galileo/sb/galileo\\_datasheet\\_329681\\_003.pdf](http://download.intel.com/support/galileo/sb/galileo_datasheet_329681_003.pdf)

[31] Atmel, “*Atmel AVR 8-bit and 32-bit Microcontrollers*”. Extraído en 12-14 desde:

<http://www.atmel.com/products/microcontrollers/avr/>

[32] Alessandro D’ Ausilio, “*Arduino: A low-cost multipurpose lab equipment*”.

**Disponible en:**

<http://www.ncbi.nlm.nih.gov/pubmed/22037977>

[33] Ladyada, “*PIC vs AVR*”. Extraído en 10-14 desde:

<http://www.ladyada.net/library/picvsavr.html>

[34] Cockerell Pete, “*Arm Assembly Language Programming*”.

**Disponible en:**

<http://www.peter-cockerell.net/aalp/html/frames.html>

[35, 36] Cooking Hacks, “*GPRS/GSM Quadband Module for Arduino Tutorial (SIM900)*”. Extraído en 10-14, desde:

<http://www.cooking-hacks.com/documentation/tutorials/arduino-gprs-gsm-quadband-sim900>

[37] Sim Communications, “*SIM908 Hardware Design*”. Manual Técnico, 2011.

**Disponible en:**

<http://www.datasheetspdf.com/PDF/SIM908-C/771596/1>

[38] ITU – T, 07/2003. “*SERIES V: DATA COMMUNICATION OVER THE TELEPHONE NETWORK*”. Versión 2.50. Apéndice I, Summary of extended and format commands.

**Disponible en:**

<http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=6484>

[39] Cando Luis, 2011. “*Bloqueo electrónico en el encendido de un vehículo, para brindar un sistema de seguridad contra robos*”. Repositorio Digital UTA. Tesis de grado.

**Disponible en:**

<http://repo.uta.edu.ec/handle/123456789/444>

[40] FRAX, Biometría. “*¿Para qué sirve la biometría?*”.

**Disponible en:**

[http://www.biometricos.cl/equipos\\_biometria/para\\_que\\_sirve\\_la\\_biometria\\_y\\_control\\_de\\_huella.php](http://www.biometricos.cl/equipos_biometria/para_que_sirve_la_biometria_y_control_de_huella.php)

[41, 42] Tolosa Borja – Giz Bueno, “*Sistemas Biométricos*”.

**Disponible en:**

[http://www.dsi.uclm.es/personal/MiguelFGraciani/mikicurri/Docencia/Bioinformatica/web\\_BIO/Documentacion/Trabajos/Biometria/Trabajo%20Biometria.pdf](http://www.dsi.uclm.es/personal/MiguelFGraciani/mikicurri/Docencia/Bioinformatica/web_BIO/Documentacion/Trabajos/Biometria/Trabajo%20Biometria.pdf)

[43] Daniel Tapias Merino, “*Sistemas de Reconocimiento de Voz en las Telecomunicaciones*”. Volumen Monográfico.

**Disponible en:**

[dialnet.unirioja.es/descarga/articulo/227026.pdf](http://dialnet.unirioja.es/descarga/articulo/227026.pdf)

[44] Travieso, Del Pozo, Ticay, “*Sistemas Biométricos*”. Cuaderno de red de cádras telefónicas, Universidad de las Palmas de Gran Canaria, 2011.

**Disponible en:**

[http://www.rcysostenibilidad.telefonica.com/blogs/documentoscatedras/files/2012/07/Catedra\\_telefonica\\_Sistemas\\_Biometricos.pdf](http://www.rcysostenibilidad.telefonica.com/blogs/documentoscatedras/files/2012/07/Catedra_telefonica_Sistemas_Biometricos.pdf)

[45] Barrios Cárdenas, 01/11/12. “*Sistema de Control de un Manipulador Mecánico mediante Detección de Voz*”. Reconocimiento de voz.

**Disponible en:**

<http://revistas.udistrital.edu.co/ojs/index.php/REDES/article/view/6377/7893>

[46, 47] Reyes Carlos, “*Microcontroladores PIC*”. 3<sup>ra</sup> ed, Editorial Rispergraf, 2008.

[48] BMW Motorrad, “*R1200GS Adventure Rider’s Manual (US)*”. 2014.

**Disponible en:**

[http://www.bmwmotorcycles.com/us/en/index.html?content=http://www.bmwmotorcycles.com/us/en/services/manuals/manuals\\_main.html&notrack=1](http://www.bmwmotorcycles.com/us/en/index.html?content=http://www.bmwmotorcycles.com/us/en/services/manuals/manuals_main.html&notrack=1)

[49] KTM, “*KTM Adventure 990 Owner’s Manul US*”. 2011.

**Disponible en:**

[http://ktm950.info/how/owners\\_manuals/owners\\_manuals.html](http://ktm950.info/how/owners_manuals/owners_manuals.html)

[50] Harley Davidson, “*Road Glide Special*”. Características Principales. Extraído en 04-15 desde:

[http://www.harley-davidson.com/es\\_ES/Motorcycles/road-glide-special.html#!key-features](http://www.harley-davidson.com/es_ES/Motorcycles/road-glide-special.html#!key-features)

[51] BeagleBoard, “*BeagleBone System Reference Manual Rev A5.0.0*”.

**Disponible en:**

<http://www.element14.com/community/docs/DOC-50167/1/circuitco-beaglebone-rev-a5-system-reference-manual>

[52] Raspberry Org, “*Raspberry Pi B2*”. Extraído en 03-15 desde:

<https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>

[53] Raspberry Org, “*Raspberry Pi A+*”. Extraído en 11-14 desde:



<https://www.raspberrypi.org/products/model-a-plus/>

[54] Arduino. “*Arduino Mega*”. Board Overview. Extraído en 08-14 desde:

<http://www.arduino.cc/en/pmwiki.php?n=Main/arduinoBoardMega>

[55] Gadget Factory, “*Papilio One Hardware Guide*”. Extraído en 08-14 desde:

<http://papilio.cc/index.php?n=Papilio.Hardware>

[56, 57] Arduino. “*Arduino Mega*”. Board Overview. Extraído en 08-14 desde:

<http://www.arduino.cc/en/pmwiki.php?n=Main/arduinoBoardMega>

[58, 59] Arduino. “*Arduino Uno*”. Board Overview. Extraído en 08-14 desde:

<http://www.arduino.cc/en/Main/ArduinoBoardUno>

[60] Aliexpress, “*GSM GPS shield for UNO SIM908 developments kits*”. Extraído en 08-14 desde:

[http://www.aliexpress.com/store/product/Free-shipping-GPRS-GPS-SIM908-kits-for-Arduino-GSM-GPS-Expansion-board/605000\\_1653542395.html](http://www.aliexpress.com/store/product/Free-shipping-GPRS-GPS-SIM908-kits-for-Arduino-GSM-GPS-Expansion-board/605000_1653542395.html)

[61] HW Kitchen, “*GMS-GPS Playground Shield for Arduino*”. Extraído en 08-14 desde:

<http://www.hwkitchen.com/products/gsm-gps-playground-shield-for-arduino/>

[62] Aliexpress, “*GSM GPS shield for UNO SIM908 developments kits*”. Extraído en 08-14 desde:

[http://www.aliexpress.com/store/product/Free-shipping-GPRS-GPS-SIM908-kits-for-Arduino-GSM-GPS-Expansion-board/605000\\_1653542395.html](http://www.aliexpress.com/store/product/Free-shipping-GPRS-GPS-SIM908-kits-for-Arduino-GSM-GPS-Expansion-board/605000_1653542395.html)

[63] OTTO Wirelees Solutions, “*SIM908 Datasheet*”.

**Disponible en:**

<http://www.dfrobot.com/image/data/TEL0051/3.0/SIM908%20datasheet.pdf>

[64] Sim Communications, “*SIM 908 Hardware Design*”. Manual técnico, versión 1.00, 2011.

**Disponible en:**

<http://www.datasheetspdf.com/PDF/SIM908-C/771596/1>

[65] VEEAR, “*EasyVR User Manual*”. Versión 3.6.7.

**Disponible en:**

[http://www.veear.eu/files/easyvr\\_user\\_manual\\_3.6.7.pdf](http://www.veear.eu/files/easyvr_user_manual_3.6.7.pdf)

[66] Roboard, “*RM-G212 16X4 Thermal Array Modulo*”. Extraído en 09-14 desde:  
<http://www.roboard.com/G212.html>

[67] VEEAR, “*Easy VR 2.0*”. Easy VR Shield Overview. Extraído en 09-14 desde:  
<http://www.veear.eu/products/old-products/easyvr/>

[68-71] VEEAR, “*EasyVR User Manual*”. Manual de usuario, versión 3.6.7.

**Disponible en:**

[http://www.veear.eu/files/easyvr\\_user\\_manual\\_3.6.7.pdf](http://www.veear.eu/files/easyvr_user_manual_3.6.7.pdf)

[72] Reyes Carlos, “*Microcontroladores PIC*”. 3<sup>ra</sup> ed, Editorial Rispergraf, 2008.

[73] Sparkfun Electronics, “*Graphic LCD 84x48 - Nokia 5110*”. Extraído en 08-14 desde:

<https://www.sparkfun.com/products/10168>

[74] Sparkfun Electronics. “*Graphic LCD 84x48 - Nokia 5110*”. Extraído en 08-14 desde:

<https://www.sparkfun.com/products/10168>

[75] Karlsen Henning, “*Library: LCD5110\_Basic*”. Extraído en 10-14 desde:

<http://www.rinkydinkelectronics.com/library.php?id=45>

[76] Bajaj, “*Pulsar DTS-i UG-III-180cc Training Notes*”. Manual de Reparaciones, 1<sup>ra</sup> ed.

[77,78] Arias Paz M, “*Motocicletas*”. 32<sup>va</sup> ed, Inversiones Editoriales Dossat, 2003.

[79] Bajaj, “*Pulsar DTS-i UG-III-180cc Training Notes*”. Manual de Reparaciones, 1<sup>ra</sup> ed.

[80] Arias Paz M, “*Motocicletas*”. 32<sup>va</sup> ed, Inversiones Editoriales Dossat, 2003.

[81] IDAE España, “*Manual de la conducción eficiente*”.

**Disponible en:**

[http://www.idae.es/uploads/documentos/documentos\\_manualPME\\_6bc54e20.pdf](http://www.idae.es/uploads/documentos/documentos_manualPME_6bc54e20.pdf)

[82,83] Bajaj, “*Pulsar DTS-i UG-III-180cc Training Notes*”. Manual de Reparaciones, 1<sup>ra</sup> ed.

# **ANEXOS**

# **ANEXO A**

## **CÓDIGO FUENTE**

# A1: BLOQUE DE SEGURIDAD: MÓDULO GSM GPS

```
#include "SIM900.h"
#include <string.h>
#include <SoftwareSerial.h>
#include <LCD5110_Basic.h>
//#include "inetGSM.h"
#include "sms.h"
#include "gps.h"
#include "call.h"
GPSSMS gps;
SMSGSM sms;
CallGSM call;

//pin2=tx,pin3=rx

int set=8,reset=9,hw_check=13,d,mod=0,sat,spd;
boolean st=false;
char fwsms[160],n[20];
char own["+593983061876",
pass["toor",en1["+593999831145",en2["+593983599135",en3["+593983303587",num[10];
char x[20],y[20],z[20],v[20],t[20],pos[150];
char pack[20],cmd[2];

void setup(){
  pinMode(set,OUTPUT);
  pinMode(reset,OUTPUT);
  pinMode(hw_check,OUTPUT);
  randomSeed(analogRead(0));
  Serial.begin(9600);
  Serial.println("BAGHEERA GSM/GPS IS STARTING...");
  if(gsm.begin(2400)){
    Serial.println("BAGHEERA GSM/GPS ONLINE");
    st=true;
  }
  else{
    Serial.println("BAGHEERA CAN NOT REACH SEVER");
  }
}

int check_line(char *n){
  int spd=strcmp(n,own);
  return spd;
}

void mainsw(int mod,char *n){
  int spd;
  switch (mod){
    case 1:
      Serial.println("Incoming Request for STATUS");
      sms.SendSMS(own,"System Current Status: ONLINE --
BAGHEERA");
      Serial.println("Done");
      break;
```

```

case 2:
    Serial.println("Incoming Request for: MODEL");
    sms.SendsSMS(own,"MOD: FZ 2014, HOUSE: Yamaha, ENGINE:
180cc, NUMBER: HX023H -- BAGHEERA");
    Serial.println("Done");
break;
case 3:
    Serial.println("Incoming Request for: POSITION");
    //gsm.forceON;
    for(int j=0; j<5;j++){
        if (gps.attachGPS())
        {
            Serial.println("*BAGHHERA GPS IS ONLINE");
        }
        else{
            Serial.println("*BAGHEERA GPS IS OFFLINE");
            sms.SendsSMS(own,"GPS not available -- BAGHEERA");
            break;
        }
        delay (22000);
        sat=gps.getStat();
    }
    gps.getPar(x,y,z,t,v);
    if(sat==3)
    {
        strcat(pos,"Latitude: ");
        strcat(pos,x);
        strcat(pos," ;Longitude: ");
        strcat(pos,y);
        strcat(pos," ;Height: ");
        strcat(pos,z);
        strcat(pos," ;Speed:");
        strcat(pos,v);
        strcat(pos,"-- BAGHEERA");
        Serial.println("Done");
        sms.SendsSMS(own,pos);
        break;
    }
    else{
        sms.SendsSMS(own,"Position not available -- BAGHEERA");
    }
break;
case 4:
    Serial.println("Incoming Request for: MANUAL");
    sms.SendsSMS(own,"MOD: vehicle info, STA: system info, POS:
use GPS, MAN: help -- BAGHEERA");
    Serial.println("Done");
    break;
case 5:
    Serial.println("Incoming Request for: SOS DECOY");
    for(int j=0; j<3; j++){
        call.Call(own);
        delay (8000);
        call.HangUp();
        delay (250);
    }
    Serial.println("Done");
break;

```

```

    case 6:
        Serial.println("Incoming Request for: SOS Broadcast");
        sms.SendsSMS(own,"I'm in trouble, please call me:
0983061876 -- BAGHEERA");
        Serial.println("Done");
        break;
    case 7:
        Serial.println("Incoming Request for: SWITCH ENGAGE");
        for(int j=0; j<4; j++){
            if(pass[j]!=n[j+4]){
                spd=0;
                break;
            }
            else{
                spd=1;
                continue;
            }
        }
        if(!spd)
            Serial.println("Invalid Password");
        else{
            digitalWrite(set,HIGH);
            delay(1500);
            digitalWrite(set,LOW);
            sms.SendsSMS(own,"Emergency Switch is now engaged --
BAGHEERA");
        }
        Serial.println("Done");
        break;
    case 8:
        Serial.println("Incoming Request for: SWITCH DISENGAGE");
        for(int j=0; j<4; j++){
            if(pass[j]!=n[j+4]){
                spd=0;
                break;
            }
            else{
                spd=1;
                continue;
            }
        }
        if(!spd)
            Serial.println("Invalid Password");
        else{
            digitalWrite(reset,HIGH);
            delay(1500);
            digitalWrite(reset,LOW);
            sms.SendsSMS(own,"Emergency Switch is now disengaged --
BAGHEERA");
        }
        Serial.println("Done");
        break;
    default:
        Serial.println("Incoming Request for: NA");
        sms.SendsSMS(own,"404 CMD NOT FOUND, Need some help? Try
texting MAN -- BAGHEERA");
        Serial.println("Done");
}

```

```

}

void check_gsm(){
  if(gsm.readSMS(fwsms,160,n,20)){
    cmd[0]=fwsms[0];cmd[1]=fwsms[1];cmd[2]=fwsms[2];
    if(cmd[0]=='S'&&cmd[1]=='T'&&cmd[2]=='A')
      mod=1;
    else if(cmd[0]=='M'&&cmd[1]=='O'&&cmd[2]=='D')
      mod=2;
    else if(cmd[0]=='P'&&cmd[1]=='O'&&cmd[2]=='S')
      mod=3;
    else if(cmd[0]=='M'&&cmd[1]=='A'&&cmd[2]=='N')
      mod=4;
    else if(cmd[0]=='D'&&cmd[1]=='E'&&cmd[2]=='C')
      mod=5;
    else if(cmd[0]=='B'&&cmd[1]=='R'&&cmd[2]=='D')
      mod=6;
    else if(cmd[0]=='P'&&cmd[1]=='O'&&cmd[2]=='W')
      mod=7;
    else if(cmd[0]=='H'&&cmd[1]=='L'&&cmd[2]=='T')
      mod=8;
    if(check_line(n))
      Serial.println("Not authorized.");
    else{
      Serial.println("Authorized");
      mainsw(mod,n);
    }
    for(int j=0; j<20;j++)
      fwsms[j]=n[j]=0;
  }
}

void loop(){
  check_gsm();
}

```



## A2: BLOQUE DE SEGURIDAD: MÓDULO DE RECONOCIMIENTO DE VOZ

```
#if defined(ARDUINO) && ARDUINO >= 100
#include "Arduino.h"
#include "Platform.h"
#include "SoftwareSerial.h"
#ifndef CDC_ENABLED
SoftwareSerial port(12,13);
#else
#define port Serial1
#endif
#else
#include "WProgram.h"
#include "NewSoftSerial.h"
NewSoftSerial port(12,13);
#endif

#include "EasyVR.h"
EasyVR easyvr(port);

enum Groups
{
    GROUP_1 = 1,
};

enum Group::TRIGGER
{
    G0_SOY_YO = 0
}
enum Group1
{
    G1_EN_LINEA = 0,
    G1_ARRANCA = 1,
    G1_BLOQUEATE = 2,
    G1_ASISTEME = 3,
    G1_STATUS = 4;
    G1_ME_VOY = 5,
};

EasyVRBridge bridge;
int8_t group,idx;
int j=0,i=0,drv=0,block=0;
int set=8,reset=9,est=10;

void setup(){
    pinMode(set,OUTPUT);
    pinMode(reset,OUTPUT);
    pinMode(est,OUTPUT);
    #ifndef CDC_ENABLED
    if (bridge.check()){
        cli();
        bridge.loop(0, 1, 12, 13);
    }
    Serial.begin(9600);
}
```

```

Serial.println("Bridge not started!");
#else
if (bridge.check()){
    port.begin(9600);
    bridge.loop(port);
}
Serial.println("Bridge connection aborted!");
#endif
port.begin(9600);
while (!easyvr.detect()){
    Serial.println("EasyVR not detected!");
    delay(1000);
}
easyvr.setPinOutput(EasyVR::IO1, LOW);
Serial.println("EasyVR detected!");
easyvr.setTimeout(5);
easyvr.setLanguage(4);
group=1;
}

void execute()
{
    switch (group){
        case GROUP_1:
            switch (idx){
                case G1_EN_LINEA:
                    if(!(block>0)){
                        easyvr.playSound(11, EasyVR::VOL_FULL);
                        delay(50);
                        easyvr.playSound(03, EasyVR::VOL_FULL);
                        delay(50);
                        block=1;
                        digitalWrite(set, HIGH);
                        delay(500);
                        digitalWrite(set, LOW);
                    }
                else{
                    easyvr.playSound(12, EasyVR::VOL_FULL);
                }
            }
            break;
        case G1_ARRANCA:
            if(!block){
                easyvr.playSound(12, EasyVR::VOL_FULL);
                delay(50);
                easyvr.playSound(6, EasyVR::VOL_FULL);
            }
            else{
                easyvr.playSound(11, EasyVR::VOL_FULL);
                delay(50);
                easyvr.playSound(8, EasyVR::VOL_FULL);
                digitalWrite(est, HIGH);
                delay(3200);
                digitalWrite(est, LOW);
            }
            break;
        case G1_BLOQUEATE:
            if(block){
                easyvr.playSound(11, EasyVR::VOL_FULL);
            }
    }
}

```

```

        delay(50);
        easyvr.playSound(04, EasyVR::VOL_FULL);
        block=0;
        digitalWrite(reset, HIGH);
        delay(500);
        digitalWrite(reset, LOW);
    }
    else{
        easyvr.playSound(12, EasyVR::VOL_FULL);
        delay(50);
    }
    break;
    case G1_STATUS:
        easyvr.playSound(11, EasyVR::VOL_FULL);
        delay(50);
        easyvr.playSound(13, EasyVR::VOL_FULL);
        break;
    case G1_ASISTEME:
        if(!drv&&(i>0)){
            easyvr.playSound(11, EasyVR::VOL_FULL);
            delay(50);
            easyvr.playSound(7, EasyVR::VOL_FULL);
            drv=1;
            i++;
        }
        else{
            easyvr.playSound(12, EasyVR::VOL_FULL);
            delay(50);
            easyvr.playSound(5, EasyVR::VOL_FULL);
        }
        break;
    case G1_ME_VOY:
        easyvr.playSound(12, EasyVR::VOL_FULL);
        delay(50);
        easyvr.playSound(5, EasyVR::VOL_FULL);
    }
    break;
}
break;
}
}

void vr_listen()
{
    easyvr.setPinOutput(EasyVR::IO1, HIGH); //start listening
    Serial.print("Listening...");
    Serial.println(group);
    easyvr.recognizeCommand(group);
    do{
    }while(!easyvr.hasFinished());
    easyvr.setPinOutput(EasyVR::IO1, LOW); //finish listening
    idx=easyvr.getword();
    if (idx>=0){
        return;
    }
    idx=easyvr.getCommand();
    if (idx>=0){
        uint8_t train=0;

```

```
char name[32];
Serial.print("Command: ");
Serial.print(idx);
if (easyvr.dumpCommand(group,idx,name,train)){
    Serial.print(" = ");
    Serial.println(name);
}
else
    Serial.println();
//easyvr.playSound(0, EasyVR::VOL_FULL);
execute();
}
else{
    if (easyvr.isTimeout())
        Serial.println("Timed out, try again...");
    int16_t err=easyvr.getError();
    if (err>=0){
        Serial.print("Error ");
        Serial.println(err, HEX);
    }
}
}

void loop(){
    vr_listen();
}
```

## A3: BLOQUE DE CONDUCCIÓN ASISTIDA

```
#include <LCD5110_Basic.h>
LCD5110 myGLCD(8,9,10,11,12);
const int x=5;
const int sample=1000;
int k,gear,rn;
float spd, rin=48.26;//rin 19
boolean t;

extern uint8_t SmallFont[];

uint8_t bagheera[] PROGMEM={
bagheera(bitmap);
};
uint8_t one[] PROGMEM={
one(bitmap);
};
uint8_t two[] PROGMEM={
two(bitmap);
};
uint8_t three[] PROGMEM={
three(bitmap);
};
uint8_t four[] PROGMEM={
four(bitmap);
};
uint8_t five[] PROGMEM={
five(bitmap);
};

void setup(){
  randomSeed(analogRead(0));
  myGLCD.InitLCD();
  myGLCD.setContrast(70);
  myGLCD.setFont(SmallFont);
  Serial.begin(9600);
  k=1;//number of edges
  TCCR1A=0;//setting timer1 register A=0;
}

unsigned int pulse_in(){
  unsigned int y;
  bitSet(TCCR1B,CS12);
  bitSet(TCCR1B,CS11);
  delay(sample);
  TCCR1B=0;
  y=TCNT1;
  TCNT1=0;
  return(y);
}

void drv(int gear,boolean t){
  switch(gear){
    case 0:
```

```

        Serial.println("Speed [Km/h]: ");
        Serial.println(spd);
        myGLCD.clrScr();
        myGLCD.drawBitmap(0,0,bagheera,84,48);
        myGLCD.invert(t);
    break;
    case 1:
        Serial.println("Speed [Km/h]: ");
        Serial.println(spd);
        myGLCD.clrScr();
        myGLCD.drawBitmap(0,0,one,84,48);
        myGLCD.invert(t);
    break;
    case 2:
        Serial.println("Speed [Km/h]: ");
        Serial.println(spd);
        myGLCD.clrScr();
        myGLCD.drawBitmap(0,0,two,84,48);
        myGLCD.invert(t);
    break;
    case 3:
        Serial.println("Speed [Km/h]: ");
        Serial.println(spd);
        myGLCD.clrScr();
        myGLCD.drawBitmap(0,0,three,84,48);
        myGLCD.invert(t);
    break;
    case 4:
        Serial.println("Speed [Km/h]: ");
        Serial.println(spd);
        myGLCD.clrScr();
        myGLCD.drawBitmap(0,0,four,84,48);
        myGLCD.invert(t);
    break;
    case 5:
        Serial.println("Speed [Km/h]: ");
        Serial.println(spd);
        myGLCD.clrScr();
        myGLCD.drawBitmap(0,0,five,84,48);
        myGLCD.invert(t);
    break;
    default:
    break;
}
}

void loop(){
    spd=(pulse_in()/k)*PI*0.4826;
    spd*=3.6;
    spd=ceil(spd);
    if(spd<=2)
        gear=0;
    else if(spd>2&&spd<=19)
        gear=1;
    else if(spd>19&&spd<=29)
        gear=2;
    else if(spd>29&&spd<=39)
        gear=3;
}

```

```
else if(spd>39&&spd<=49)
  gear=4;
else if(spd>49){
  if(spd<110)
    gear=5;
  else
    gear=6;
}
rn=random(0,10);
if(rn<=5)
  t=LOW;
else
  t=HIGH;
drv(gear,t);
delay(500);
}
```

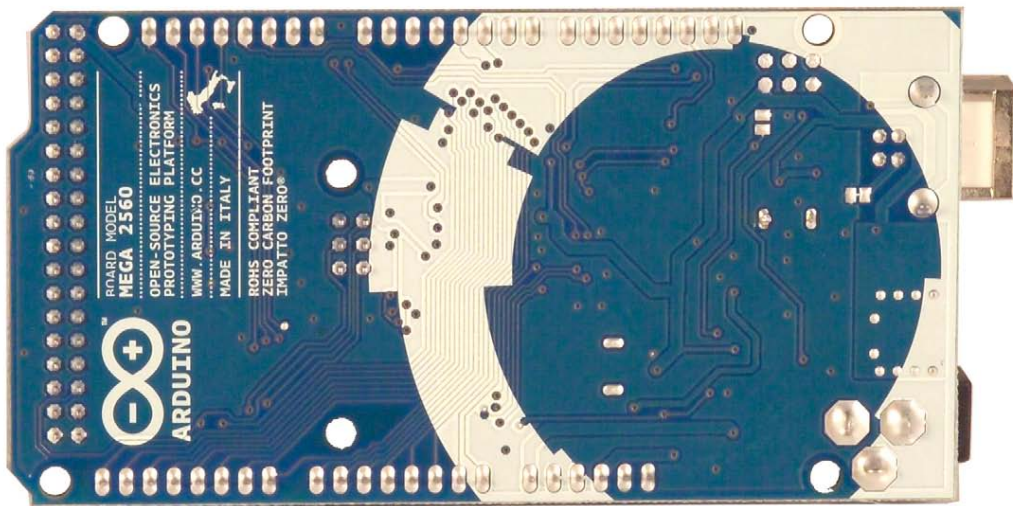
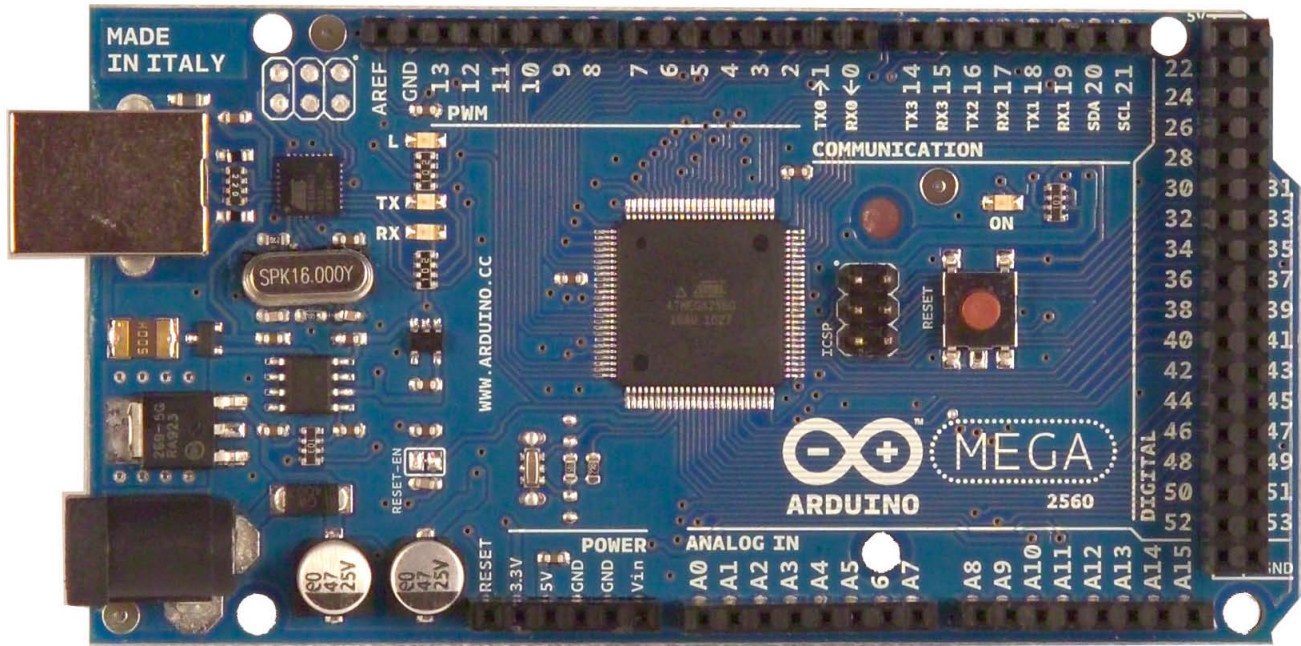
# **ANEXO B**

## **HOJAS DE DATOS**



## **B1: ARDUINO MEGA**

# Arduino Mega 2560



The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

## Schematic & Reference Design

EAGLE files: [arduino-mega2560-reference-design.zip](#)

Schematic: [arduino-mega2560-schematic.pdf](#)

## Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

## Power

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- + **VIN**. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- + **5V**. The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- + **3V3**. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- + **GND**. Ground pins.

## Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

## Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- + **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX)**. Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- + **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2)**. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- + **PWM: 0 to 13**. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- + **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS)**. These pins support SPI communication using the [SPI library](#). The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.
- + **LED: 13**. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- + **I<sup>2</sup>C: 20 (SDA) and 21 (SCL)**. Support I<sup>2</sup>C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I<sup>2</sup>C pins on the Duemilanove or Diecimila.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and [analogReference\(\)](#) function.

There are a couple of other pins on the board:

✦ **AREF**. Reference voltage for the analog inputs. Used with `analogReference()`.

✦ **Reset**. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

## Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Mega2560's digital pins.

The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation on the Wiring website for details. For SPI communication, use the SPI library.

## Programming

The Arduino Mega can be programmed with the Arduino software (download). For details, see the reference and tutorials.

The ATmega2560 on the Arduino Mega comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

The ATmega8U2 firmware source code is available in the Arduino repository. The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See this user-contributed tutorial for more information.

## Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can

have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega2560 contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

### USB Overcurrent Protection

The Arduino Mega2560 has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

### Physical Characteristics and Shield Compatibility

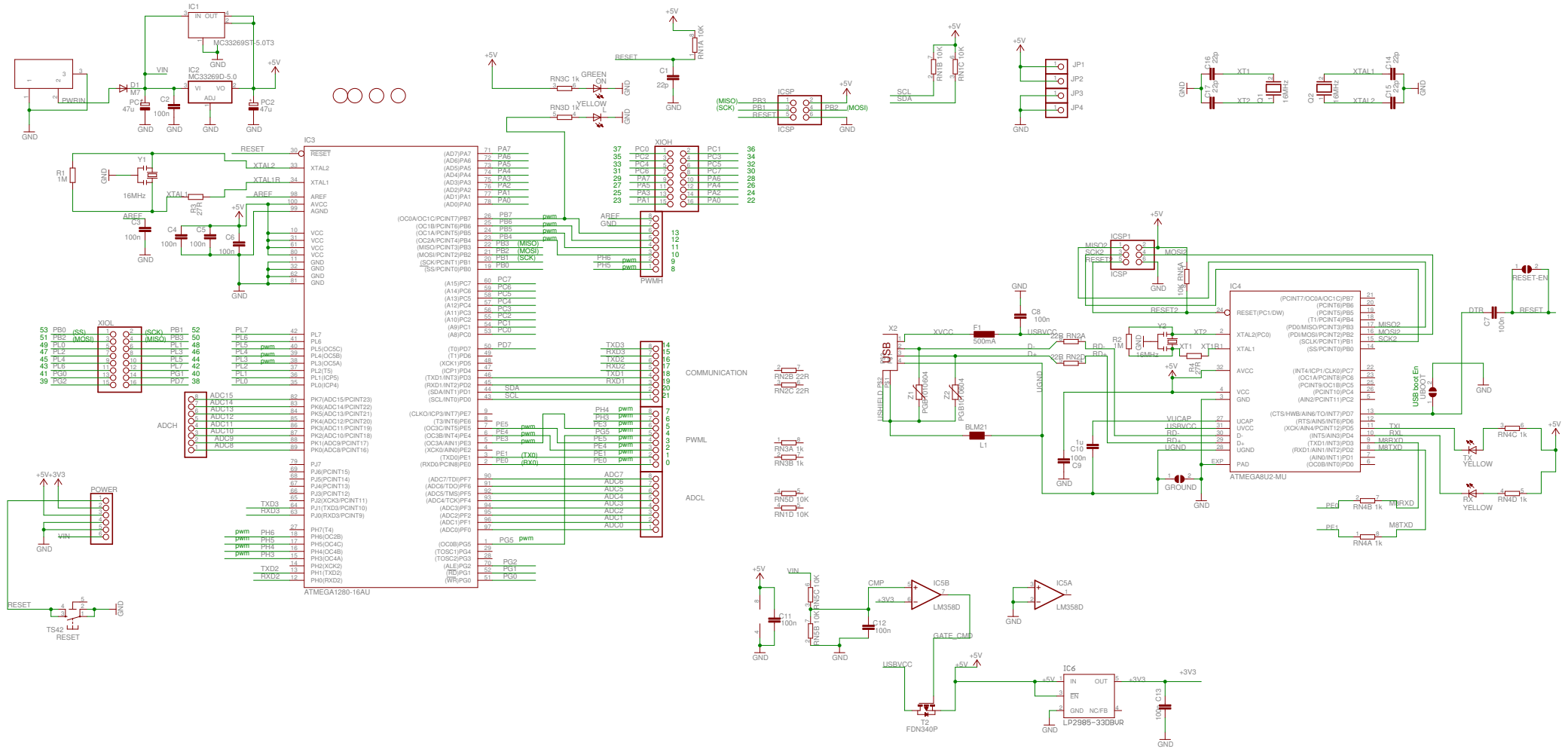
The maximum length and width of the Mega2560 PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega2560 is designed to be compatible with most shields designed for the Uno, Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega2560 and Duemilanove / Diecimila. *Please note that I<sup>2</sup>C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).*

# Arduino™ Mega 2560 Reference Design

Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS". Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

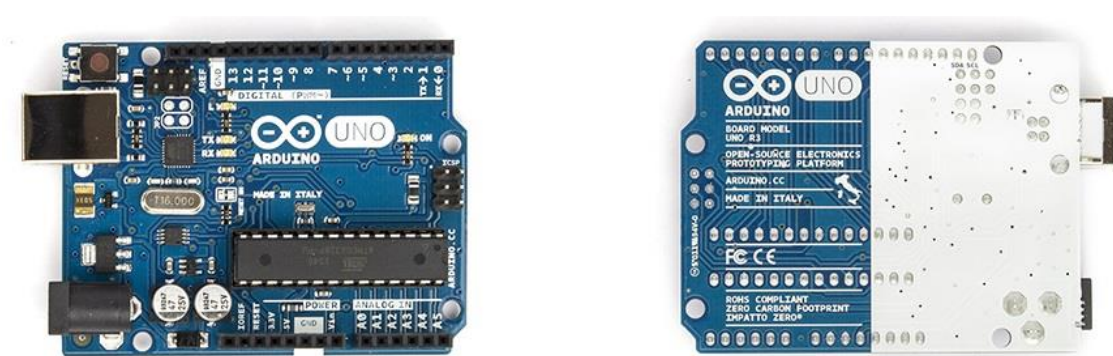
Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.



## **B2: ARDUINO UNO**



# Arduino – ArduinoBoardUno



## Overview

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

Rev 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode. Rev 3 of the board has the following new features:

- 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible with both the board that uses the AVR, which operates with 5V and with the Arduino Due that operates with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward.

## Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

## Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

- **IOREF.** This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

## Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

## Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using `pinMode()`, `digitalWrite()`, and `digitalRead()` functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the `attachInterrupt()` function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the `analogWrite()` function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the SPI library.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the `analogReference()` function. Additionally, some pins have specialized functionality:

- **TWI: A4 or SDA pin and A5 or SCL pin.** Support TWI communication using the Wire library.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with `analogReference()`.
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

## Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V)

serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Uno's digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication.

## **Programming**

The Arduino Uno can be programmed with the Arduino software. Select "Arduino Uno from the **Tools > Board** menu (according to the microcontroller on your board).

The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP or similar.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available . The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader).

## **Automatic (Software) Reset**

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line.

## **USB Overcurrent Protection**

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

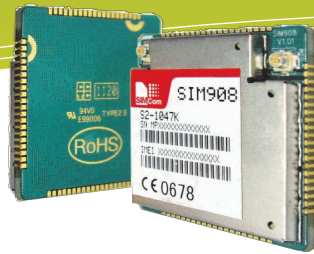
## **Physical Characteristics**

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

**B3: SIM908**

# SIM908

Combination GSM/GPRS+GPS Module in an SMT package



## OVERVIEW

The SIM908 is the first member of a new family of ARM926EJ-S based modules doubling the core performance frequency versus traditional ARM7 based solutions. At the same time, with advanced, innovative design, the SIM908 can reduce sleep mode power consumption, providing a power saving of up to 40% compared to current industry averages.

It is delivered in an SMT package size of 30x30x3.2mm.

The product features an integrated internal GPS.

Considering the high performance, low power, and small size, SIM908 is an ideal module for many M2M applications where positioning is required at an affordable price.

## GENERAL FEATURES

- Quad Band 850 / 900 / 1800 / 1900 MHz
- GPRS Multi-slot class 10
- GPRS Mobile Station class B
- Compliant to GSM phase 2 / 2+
  - Class 4 (2W @ 850 / 900 MHz)
  - Class 1 (1W @ 1800 / 1900 MHz)
- Dimensions: 30 x 30 x 3.2mm
- Weight: 5.2g
  - SIM908-C:11.1g
- Control via AT Commands (GSM 07.07, 07.05 and SIMCOM enhanced AT Commands)
- SIM Application Toolkit
- Supply Voltage Range
  - GPRS: 3.2V ~ 4.8V
  - GPS: 3.0V ~ 4.5V
- Low Power Consumption
- Normal Operating temperature: -40°C to +85°C

## Specifications for SMS via GSM/GPRS

- Point-to-point MO and MT
- SMS cell Broadcast
- Text and PDU mode

## Specifications For Audio

- Tricodec
  - Half rate (HR)
  - Full rate (FR)
  - Enhanced Full rate (EFR)
- Hand-free operation
- Echo cancelation

## Compatibility

- AT cellular command interface

## Certifications

- CE
- ROHS

## Specifications for data transfer

- GPRS Class 8/10: max 85.6 kbps downlink
- PBCCH support
- Coding Schemes CS 1, 2, 3, 4
- CSD up to 14.4 kbps
- USSD
- Non Transparent Mode
- PPP-stack
- Integrated TCP/IP stack

## Specification for GPS

- Receiver type
  - 42-channel
  - GPS L1 C/A code
  - High-performance STE engine
- Sensitivity
  - Tracking: -160 dBm
  - Cold starts: -143 dBm
- Time-To-First-Fix
  - Cold starts: 30s (typ.)
  - Hot starts: 1s (typ.)
- Accuracy
  - Horizontal position: <2.5m CEP
- Power consumption (GSM engine in idle mode)
  - Acquisition 77mA
  - Tracking 76mA

## Interface

- 80-pad with SMT type
- Interface to external SIM 3V/1.8V
- Dual analog audio interface
- SPI interface
- RTC backup
- Charge interface
- A serial interface and a debug Interface for GSM/GPRS
- Debug interface for GPS NMEA Information output
- Two separate antenna connectors for GSM /GPRS & GPS



## **B4: EASYVR 2.0**

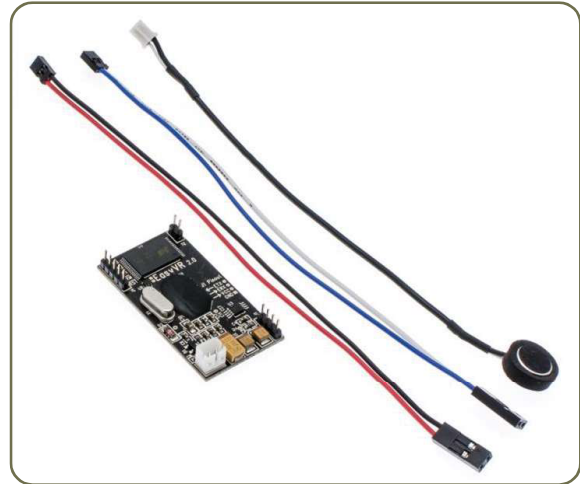


## EasyVR Module

### Product Description

EasyVR 2.0 is a multi-purpose speech recognition module designed to easily add versatile, robust and cost effective speech recognition capabilities to almost any application.

The EasyVR 2.0 module can be used with any host with an UART interface powered at 3.3V – 5V, such as PIC and Arduino boards. Some application examples include home automation, such as voice controlled light switches, locks, curtains or kitchen appliances, or adding “hearing” to the most popular robots on the market.



### EasyVR 2.0 Features

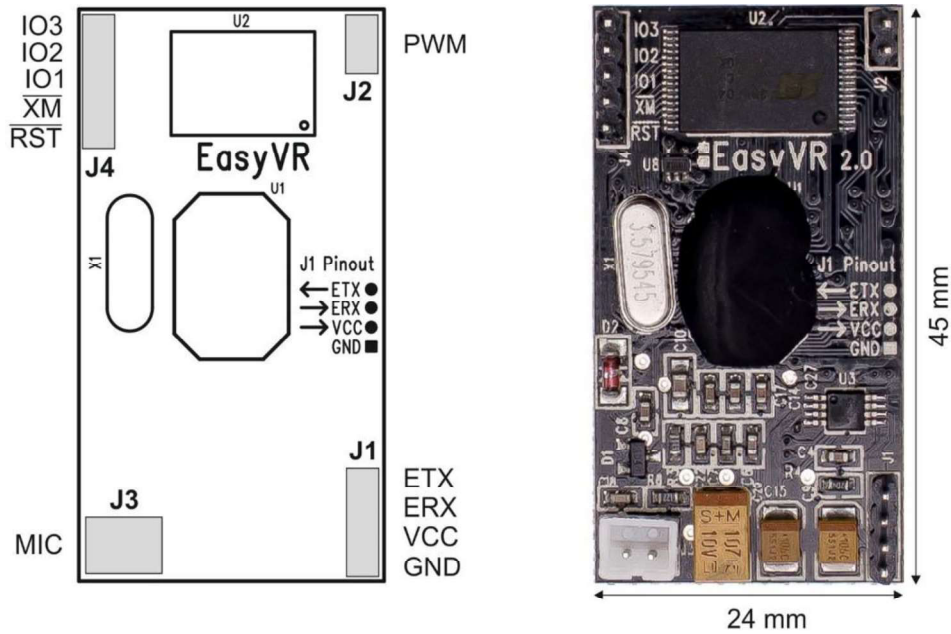
- Supports up to 28 custom Speaker Independent (SI) command vocabularies<sup>1,2</sup> (new to EasyVR 2.0)  
Supported Languages:
  - US English
  - UK English
  - German
  - French
  - Italian
  - LA Spanish
  - Korean
  - Japanese
- Supports up to 32 user-defined Speaker Dependent (SD) triggers or commands as well as Voice Passwords. SD custom commands can be spoken in ANY language.
- A variety of built-in Speaker Independent (SI) commands for ready to run basic controls, in the followings languages:
  - English (US)
  - Italian
  - German
  - French
  - Spanish
  - Japanese
- SonicNet technology for wireless communications between modules or any other sound source (Audio CD, DVD, MP3 Player)<sup>2</sup> (new to EasyVR 2.0)
- DTMF tone generation<sup>2</sup> (new to EasyVR 2.0)
- Easy-to-use and simple Graphical User Interface to program Voice Commands and audio.
- Module can be used with any host with an UART interface (powered at 3.3V - 5V)
- Simple and robust documented serial protocol to access and program through the host board
- 3 x GPIO lines (IO1, IO2, IO3) that can be controlled by new protocol commands.
- PWM audio output that directly supports 8Ω speakers.
- Sound playback of up to 9 minutes of recorded sounds or speech.

<sup>1</sup> A QuickT2SI™ Lite license (sold separately) is required to enable creation of Speaker Independent vocabularies (maximum 12 commands per set).

<sup>2</sup> Custom Speaker Independent vocabularies, SonicNet™ and DTMF generation are available since firmware version 2 (a free update from version 1 is available for download).

## Technical specifications

### Physical dimensions and pin assignment



Connector	Number	Name	Type	Description
<b>J1</b>	1	GND	-	Ground
	2	VCC	I	Voltage DC input
	3	ERX	I	Serial Data Receive (TTL level)
	4	ETX	O	Serial Data Transmit (TTL level)
<b>J2</b>	1-2	PWM	O	Differential audio output (can directly drive 8Ω speaker)
<b>J3</b>	1	MIC_RET	-	Microphone reference ground
	2	MIC_IN	I	Microphone input signal
<b>J4</b>	1	/RST	I	Active low asynchronous reset (internal 100K pull-up)
	2	/XM	I	Boot select (internal 1K pull-down)
	3	IO1	I/O	General purpose I/O ( <b>3.0 VDC</b> TTL level)
	4	IO2	I/O	General purpose I/O ( <b>3.0 VDC</b> TTL level)
	5	IO3	I/O	General purpose I/O ( <b>3.0 VDC</b> TTL level)

**Note:** the GPIO (J4.3, J4.4, and J4.5) are at nominal 3.0VDC level. Do not connect 5VDC directly to these pins!

### Recommended Operating Conditions

Symbol	Parameter	Min	Typ	Max	Unit
VCC	Voltage DC Input	3.3	5.0	5.5	V
Ta	Ambient Operating Temperature Range	0	25	70	°C
ERX	Serial Port Receive Data	0	-	VCC	V
ETX	Serial Port Transmit Data	0	-	VCC	V

### Electrical Characteristics

These are applicable to J4 pins only, including IO1-3, /XM and /RST.

Symbol	Parameter	Min	Typ	Max	Unit
V <sub>IH</sub>	Input High Voltage	2.4	3.0	3.3	V
V <sub>IL</sub>	Input Low Voltage	-0.1	0.0	0.75	V
I <sub>IL</sub>	Input Leakage Current (0 < V <sub>IO</sub> < 3V, Hi-Z Input)		<1	10	µA
R <sub>PU</sub>	Pull-up Resistance	Strong	10		kΩ
		Weak	200		kΩ
V <sub>OH</sub>	Output High Voltage (I <sub>OH</sub> = -5 mA)	2.4			V
V <sub>OL</sub>	Output Low Voltage (I <sub>OL</sub> = 8 mA)			0.6	V

### Power Supply Requirements

Symbol	Parameter	Min	Typ	Max	Unit
I <sub>Sleep</sub>	Sleep current		< 1		mA
I <sub>Oper</sub>	Operating current		12		mA
I <sub>Speaker</sub>	Audio playback current (with 8Ω speaker)		180		mA (RMS)

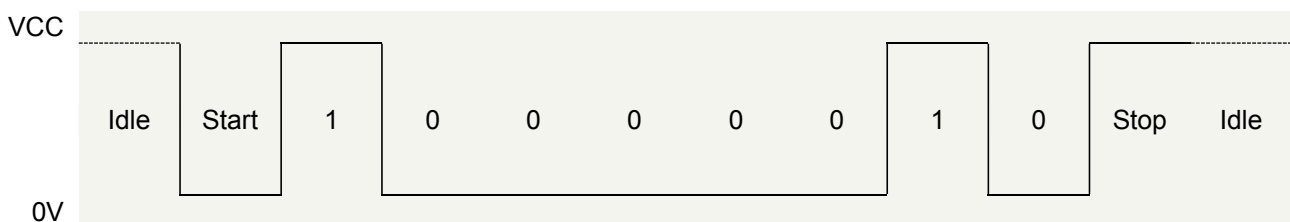
### Serial Interface

The EasyVR is a “slave” module communicating via an asynchronous serial interface (commonly known as UART interface), with the following features:

- Baud Rate: **9600** (default), 19200, 38700, 57600, 115200
- Frame: **8** Data bits, **No** parity, **1** Stop bit

The receiver input data line is ERX, while the transmitter output data line is ETX. No handshake lines are used.

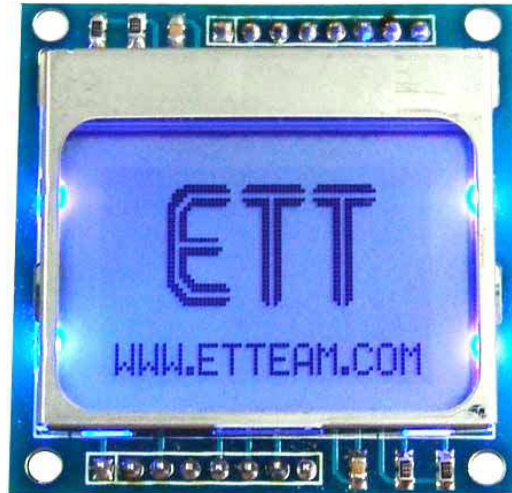
Example of a serial data frame representing character “A” (decimal 65 or hexadecimal 41):



See also chapter [Communication Protocol](#) later on this manual for communication details.

## **B5: NOKIA 5110 LCD DISPLAY**

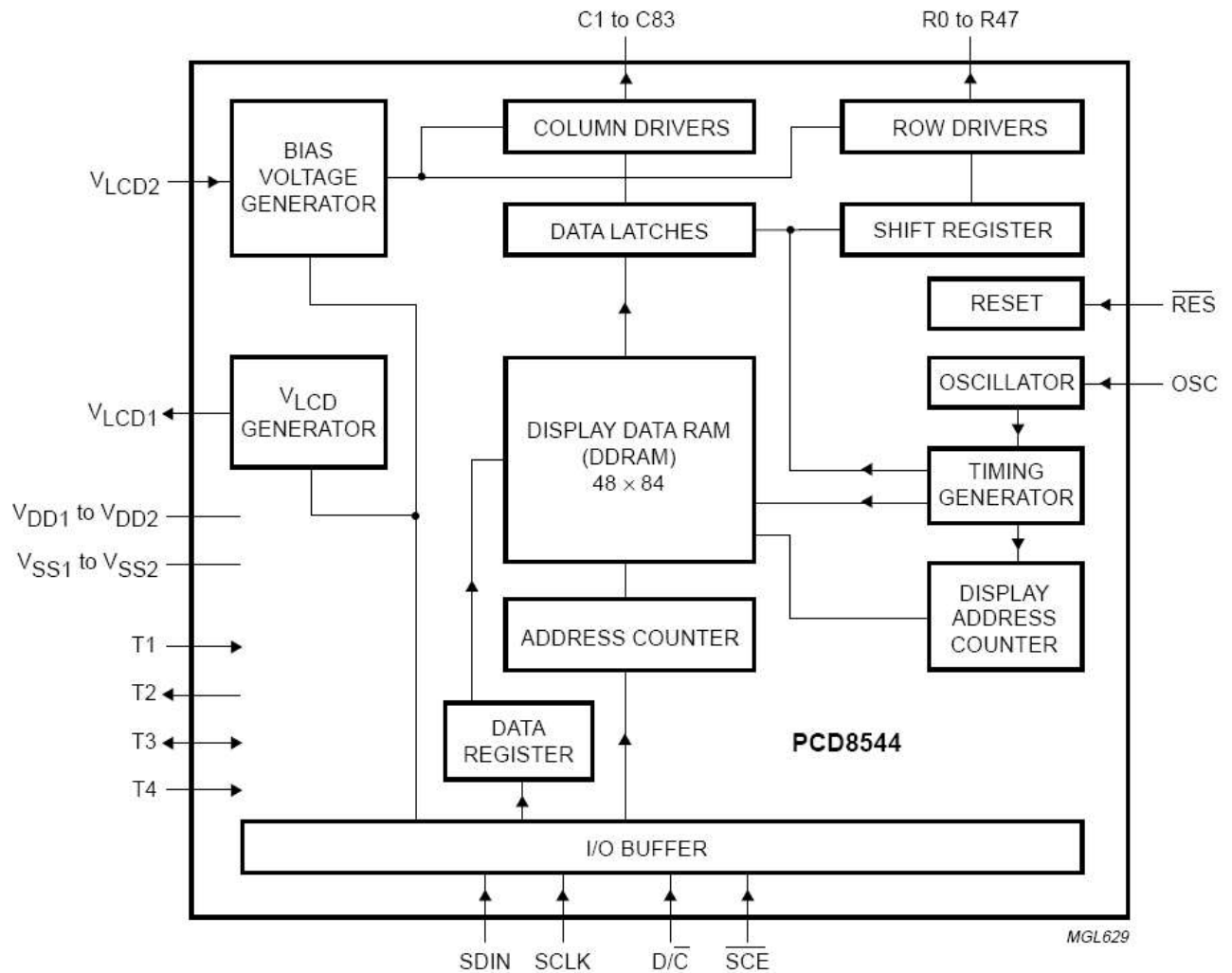
## User's Manual of Graphic LCD "ET-NOKIA LCD 5110"



### Specifications of LCD 5110

- 48 x 84 Dot LCD Display
- Serial Bus Interface with maximum high speed 4.0 Mbits/S
- Internal Controller No.PCD8544
- LED Back-Light
- Run at Voltage 2.7 -5.0 Volt
- Low power consumption; it is suitable for battery applications
- Temperature range from -25°C to +70°C
- Support Signal CMOS Input

LCD 5110 is 48 x 84 Dot LCD Graphic that has internal Controller/Driver "PCD8544" to control all displays and operations. Diagrammatic structure of internal Controller PCD8544 is shown as in figure 1.



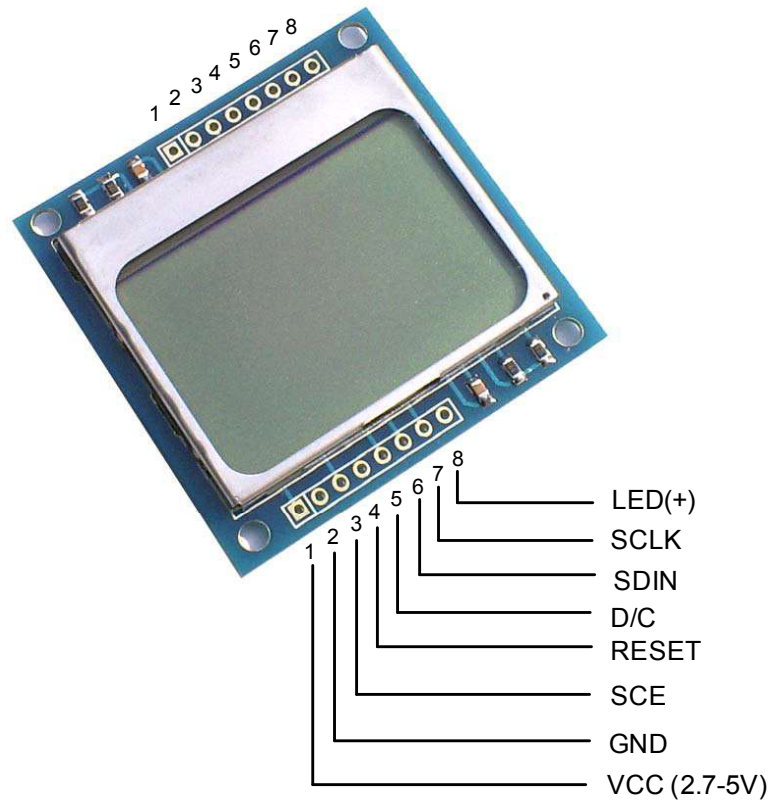
**Figure 1 shows internal structure of Controller PCD8544.**

### Handling of LCD Address (Addressing)

The address arrangement of memory that is shown on LCD Display (DDRAM) is Matrix that consists of 6 rows (Y Address) from Y-Address 0 to Y-Address 5 and 84 columns (X Address) from X-Address 0 to X-Address 83. If user wants to access to the position of displaying result on LCD Display, must refer to the relationship between X-Address and Y-Address. Data that will be sent to display is 8 bit (1 Byte) and it will be arranged as vertical line; in this case, Bit MSB will be lower and Bit LSB will be upper as shown in the following picture;

## Connection for controlling LCD Display

The signal connection for controlling operation of LCD is Serial format and there are several pins as follows;



*Figure 5 shows the pin positions of LCD.*

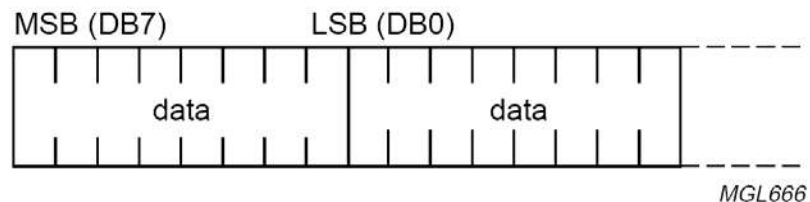
**Table 1: Function of Pin LCD**

Pin's Name	Functions
1. VCC	Pin +VCC; using Power Supply from 2.7 - 5 VCD

2. GND	Pin Ground
3. SCE	Pin CHIP ENABLE to control operation of Pin Controllers
4. RESET	Signal RESET for operation of LCD
5. D/C	Pin to configure the data formats between Data and Command.
6. SDIN	Pin DATA (SERIAL DATA LINE)
7. SCLK	Pin CLOCK (SERIAL CLOCK LINE)
8. LED	Pin to control operation of LED (Back Light)

**Communication Format**

The format of command that is used to communicate with LCD is divided into 2 modes; Command Mode and Data Mode. In this case, it uses Pin D/C to divide and control signals; if D/C = 0, the data that is sent to LCD is Command (see more detailed information of commands in the Table 1); and if D/C = 1, the data that is sent to LCD will be Data and it will placed in DDRAM Memory (Display Data RAM) to be displayed on LCD Display. After 1 byte data has already been written, 1 value of DDRAM address will be increased automatically. The format of data will be serial and it will send MSB (The Most Significant Bit) first. Generally, its structure is displayed as follows;



General format of data stream.

**Figure 6 shows the general format of Data.**

There are 2 methods to send data into LCD; firstly, sending 1 Byte data in each time and secondly, sending many continuous bytes. The Data format is shown as follows;

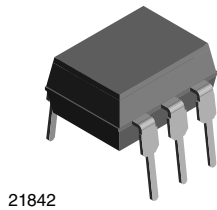
- Sending 1 byte data in each time



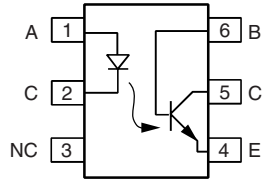
**B6: OPTOACOPLADOR 4N35**



# Optocoupler, Phototransistor Output, with Base Connection



21842



1179004-5

## DESCRIPTION

Each optocoupler consists of gallium arsenide infrared LED and a silicon NPN phototransistor.

## AGENCY APPROVALS

- Underwriters laboratory file no. E52744
- BSI: EN 60065:2002, EN 60950:2000
- FIMKO; EN 60065, EN 60335, EN 60950 certificate no. 25156

## FEATURES

- Isolation test voltage 5000 V<sub>RMS</sub>
- Interfaces with common logic families
- Input-output coupling capacitance < 0.5 pF
- Industry standard dual-in-line 6 pin package
- Compliant to RoHS directive 2002/95/EC and in accordance to WEEE 2002/96/EC



RoHS COMPLIANT

## APPLICATIONS

- AC mains detection
- Reed relay driving
- Switch mode power supply feedback
- Telephone ring detection
- Logic ground isolation
- Logic coupling with high frequency noise rejection

ORDER INFORMATION	
PART	REMARKS
4N35	CTR > 100 %, DIP-6
4N36	CTR > 100 %, DIP-6
4N37	CTR > 100 %, DIP-6

ABSOLUTE MAXIMUM RATINGS (1)				
PARAMETER	TEST CONDITION	SYMBOL	VALUE	UNIT
<b>INPUT</b>				
Reverse voltage		V <sub>R</sub>	6	V
Forward current		I <sub>F</sub>	50	mA
Surge current	t ≤ 10 μs	I <sub>FSM</sub>	1	A
Power dissipation		P <sub>diss</sub>	70	mW
<b>OUTPUT</b>				
Collector emitter breakdown voltage		V <sub>CEO</sub>	70	V
Emitter base breakdown voltage		V <sub>EBO</sub>	7	V
Collector current		I <sub>C</sub>	50	mA
	t ≤ 1 ms	I <sub>C</sub>	100	mA
Power dissipation		P <sub>diss</sub>	70	mW
<b>COUPLER</b>				
Isolation test voltage		V <sub>ISO</sub>	5000	V <sub>RMS</sub>
Creepage			≥ 7	mm
Clearance			≥ 7	mm
Isolation thickness between emitter and detector			≥ 0.4	mm

# 4N35, 4N36, 4N37



## Vishay Semiconductors Optocoupler, Phototransistor Output, with Base Connection

ABSOLUTE MAXIMUM RATINGS <sup>(1)</sup>				
PARAMETER	TEST CONDITION	SYMBOL	VALUE	UNIT
<b>COUPLER</b>				
Comparative tracking index	DIN IEC 112/VDE 0303, part 1		175	
Isolation resistance	$V_{IO} = 500 \text{ V}, T_{amb} = 25 \text{ }^\circ\text{C}$	$R_{IO}$	$10^{12}$	$\Omega$
	$V_{IO} = 500 \text{ V}, T_{amb} = 100 \text{ }^\circ\text{C}$	$R_{IO}$	$10^{11}$	$\Omega$
Storage temperature		$T_{stg}$	- 55 to + 150	$^\circ\text{C}$
Operating temperature		$T_{amb}$	- 55 to + 100	$^\circ\text{C}$
Junction temperature		$T_j$	100	$^\circ\text{C}$
Soldering temperature <sup>(2)</sup>	max. 10 s dip soldering: distance to seating plane $\geq 1.5 \text{ mm}$	$T_{sld}$	260	$^\circ\text{C}$

### Notes

<sup>(1)</sup>  $T_{amb} = 25 \text{ }^\circ\text{C}$ , unless otherwise specified.

Stresses in excess of the absolute maximum ratings can cause permanent damage to the device. Functional operation of the device is not implied at these or any other conditions in excess of those given in the operational sections of this document. Exposure to absolute maximum ratings for extended periods of the time can adversely affect reliability.

<sup>(2)</sup> Refer to wave profile for soldering conditions for through hole devices (DIP).

ELECTRICAL CHARACTERISTICS <sup>(1)</sup>								
PARAMETER	TEST CONDITION	PART	SYMBOL	MIN.	TYP.	MAX.	UNIT	
<b>INPUT</b>								
Junction capacitance	$V_R = 0 \text{ V}, f = 1 \text{ MHz}$		$C_j$		50		pF	
Forward voltage <sup>(2)</sup>	$I_F = 10 \text{ mA}$		$V_F$		1.3	1.5	V	
	$I_F = 10 \text{ mA}, T_{amb} = - 55 \text{ }^\circ\text{C}$		$V_F$	0.9	1.3	1.7	V	
Reverse current <sup>(2)</sup>	$V_R = 6 \text{ V}$		$I_R$		0.1	10	$\mu\text{A}$	
Capacitance	$V_R = 0 \text{ V}, f = 1 \text{ MHz}$		$C_O$		25		pF	
<b>OUTPUT</b>								
Collector emitter breakdown voltage <sup>(2)</sup>	$I_C = 1 \text{ mA}$	4N35	$BV_{CEO}$	30			V	
		4N36	$BV_{CEO}$	30			V	
		4N37	$BV_{CEO}$	30			V	
Emitter collector breakdown voltage <sup>(2)</sup>	$I_E = 100 \text{ } \mu\text{A}$		$BV_{ECO}$	7			V	
<b>OUTPUT</b>								
Collector base breakdown voltage <sup>(2)</sup>	$I_C = 100 \text{ } \mu\text{A}, I_B = 1 \text{ } \mu\text{A}$	4N35	$BV_{CBO}$	70			V	
		4N36	$BV_{CBO}$	70			V	
		4N37	$BV_{CBO}$	70			V	
Collector emitter leakage current <sup>(2)</sup>	$V_{CE} = 10 \text{ V}, I_F = 0$	4N35	$I_{CEO}$		5	50	nA	
		4N36	$I_{CEO}$		5	50	nA	
		4N37	$I_{CEO}$		5	50	nA	
	$V_{CE} = 30 \text{ V}, I_F = 0,$ $T_{amb} = 100 \text{ }^\circ\text{C}$	4N35	$I_{CEO}$				500	$\mu\text{A}$
		4N36	$I_{CEO}$				500	$\mu\text{A}$
		4N37	$I_{CEO}$				500	$\mu\text{A}$
Collector emitter capacitance	$V_{CE} = 0$		$C_{CE}$		6		pF	
<b>COUPLER</b>								
Resistance, input output <sup>(2)</sup>	$V_{IO} = 500 \text{ V}$		$R_{IO}$	$10^{11}$			$\Omega$	
Capacitance, input output	$f = 1 \text{ MHz}$		$C_{IO}$		0.6		pF	

### Notes

<sup>(1)</sup>  $T_{amb} = 25 \text{ }^\circ\text{C}$ , unless otherwise specified.

Minimum and maximum values are testing requirements. Typical values are characteristics of the device and are the result of engineering evaluation. Typical values are for information only and are not part of the testing requirements.

<sup>(2)</sup> Indicates JEDEC registered value.



Optocoupler, Phototransistor Output, Vishay Semiconductors with Base Connection

CURRENT TRANSFER RATIO							
PARAMETER	TEST CONDITION	PART	SYMBOL	MIN	TYP.	MAX	UNIT
DC current transfer ratio <sup>(1)</sup>	$V_{CE} = 10\text{ V}, I_F = 10\text{ mA}$	4N35	$CTR_{DC}$	100			%
		4N36	$CTR_{DC}$	100			%
		4N37	$CTR_{DC}$	100			%
	$V_{CE} = 10\text{ V}, I_F = 10\text{ mA}, T_A = -55\text{ }^\circ\text{C to } +100\text{ }^\circ\text{C}$	4N35	$CTR_{DC}$	40	50		%
		4N36	$CTR_{DC}$	40	50		%
		4N37	$CTR_{DC}$	40	50		%

Note

(1) Indicates JEDEC registered values.

SWITCHING CHARACTERISTICS							
PARAMETER	TEST CONDITION	SYMBOL	MIN.	TYP.	MAX.	UNIT	
Switching time <sup>(1)</sup>	$V_{CC} = 10\text{ V}, I_C = 2\text{ mA}, R_L = 100\text{ }\Omega$	$t_{on}, t_{off}$		10		$\mu\text{s}$	

Note

(1) Indicates JEDEC registered values.

TYPICAL CHARACTERISTICS

$T_{amb} = 25\text{ }^\circ\text{C}$ , unless otherwise specified

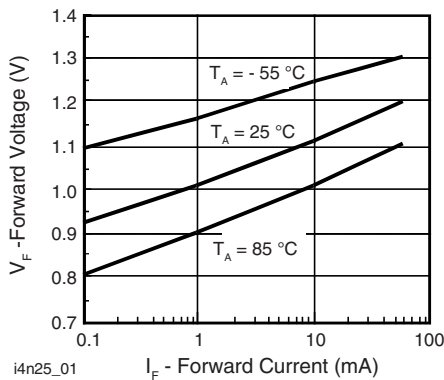


Fig. 1 - Forward Voltage vs. Forward Current

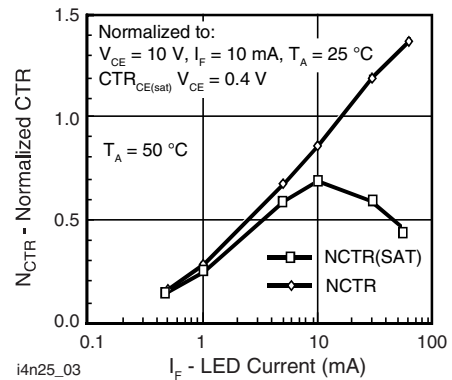


Fig. 3 - Normalized Non-Saturated and Saturated CTR vs. LED Current

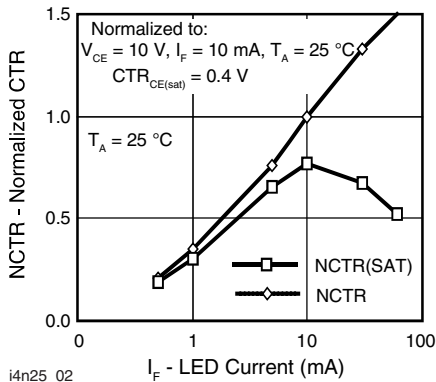


Fig. 2 - Normalized Non-Saturated and Saturated CTR vs. LED Current

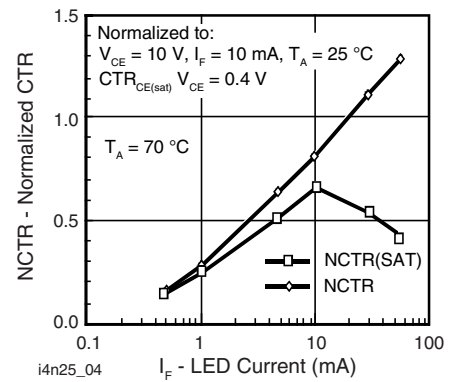
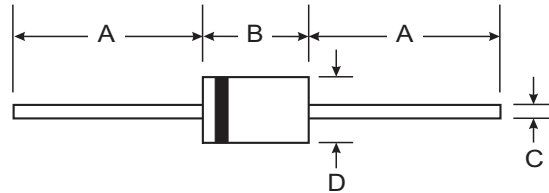


Fig. 4 - Normalized Non-Saturated and Saturated CTR vs. LED Current

**B7: DIODO 1N5408**

### Features

- Diffused Junction
- High Current Capability and Low Forward Voltage Drop
- Surge Overload Rating to 200A Peak
- Low Reverse Leakage Current
- **Lead Free Finish, RoHS Compliant (Note 3)**



### Mechanical Data

- Case: DO-201AD
- Case Material: Molded Plastic. UL Flammability Classification Rating 94V-0
- Moisture Sensitivity: Level 1 per J-STD-020C
- Terminals: Finish — Tin. Plated Leads Solderable per MIL-STD-202, Method 208 (e3)
- Polarity: Cathode Band
- Marking: Type Number
- Weight: 1.1 grams (approximate)

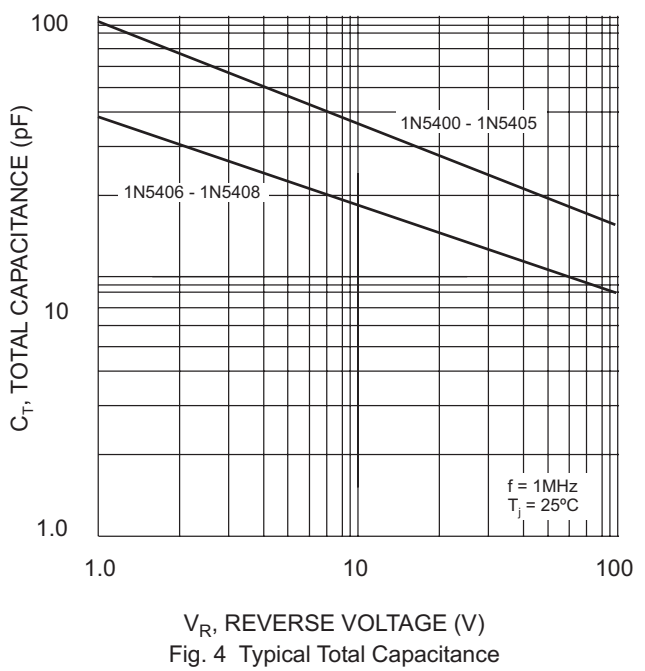
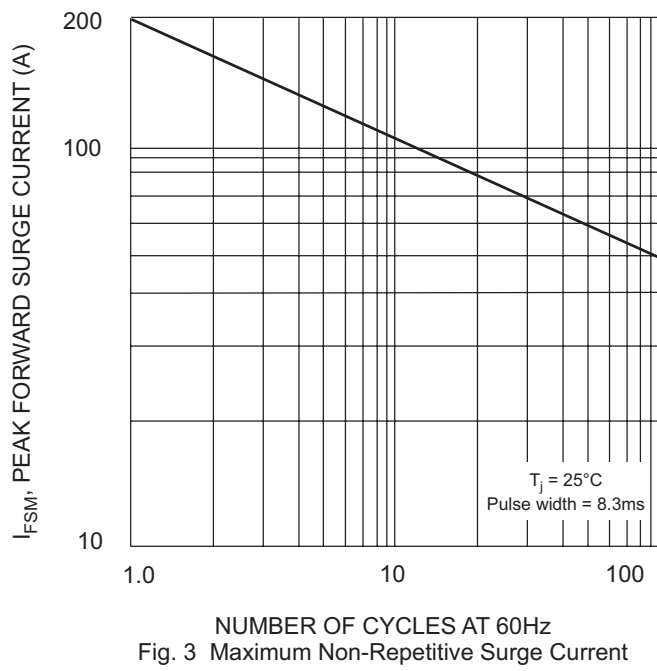
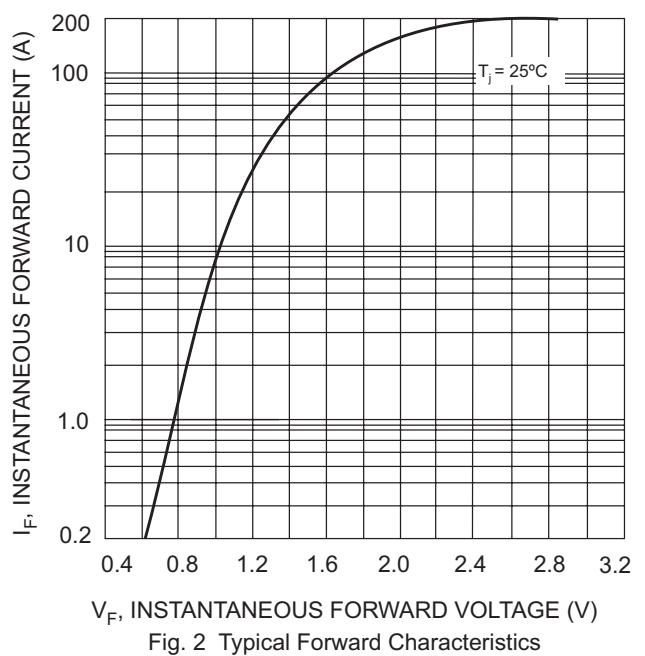
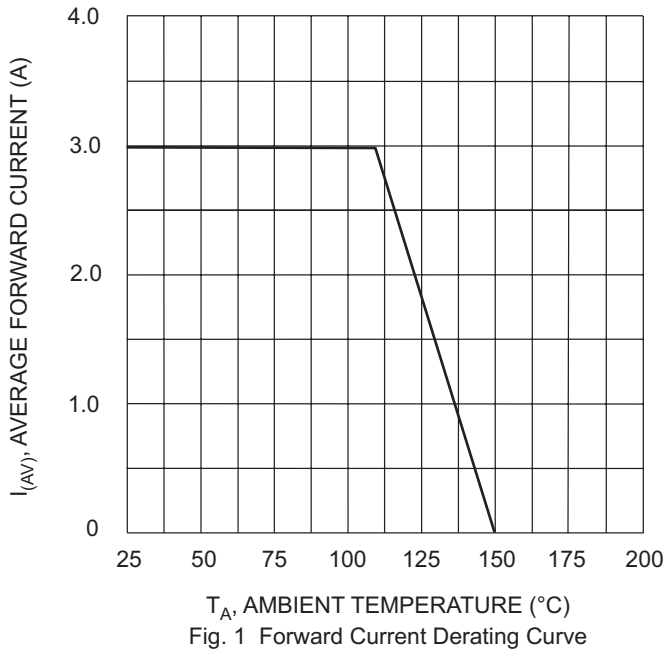
DO-201AD		
Dim	Min	Max
A	25.40	—
B	7.20	9.50
C	1.20	1.30
D	4.80	5.30
All Dimensions in mm		

### Maximum Ratings and Electrical Characteristics @ T<sub>A</sub> = 25°C unless otherwise specified

Single phase, half wave, 60Hz, resistive or inductive load.  
For capacitive load, derate current by 20%.

Characteristic	Symbol	1N 5400	1N 5401	1N 5402	1N 5404	1N 5406	1N 5407	1N 5408	Unit
Peak Repetitive Reverse Voltage Working Peak Reverse Voltage DC Blocking Voltage	V <sub>RRM</sub> V <sub>RWM</sub> V <sub>R</sub>	50	100	200	400	600	800	1000	V
RMS Reverse Voltage	V <sub>R(RMS)</sub>	35	70	140	280	420	560	700	V
Average Rectified Output Current @ T <sub>A</sub> = 105°C (Note 1)	I <sub>O</sub>	3.0							A
Non-Repetitive Peak Forward Surge Current 8.3ms Single half sine-wave superimposed on rated load	I <sub>FSM</sub>	200							A
Forward Voltage @ I <sub>F</sub> = 3.0A	V <sub>FM</sub>	1.0							V
Peak Reverse Current @ T <sub>A</sub> = 25°C at Rated DC Blocking Voltage @ T <sub>A</sub> = 150°C	I <sub>RM</sub>	10 100							μA
Typical Total Capacitance (Note 2)	C <sub>T</sub>	50				25			pF
Typical Thermal Resistance Junction to Ambient	R <sub>θJA</sub>	15							°C/W
Operating and Storage Temperature Range	T <sub>j</sub> , T <sub>STG</sub>	-65 to +150							°C

- Notes:
1. Valid provided that leads are kept at ambient temperature at a distance of 9.5mm from the case.
  2. Measured at 1.0MHz and applied reverse voltage of 4.0V DC.
  3. RoHS revision 13.2.2003. Glass and High Temperature Solder Exemptions Applied, see *EU Directive Annex Notes 5 and 7*.



**Ordering Information** (Note 4)

Device	Packaging	Shipping
1N5400-B	DO-201AD	500 Bulk
1N5400-T	DO-201AD	1.2K/Tape & Reel, 13 inch
1N5401-B	DO-201AD	500 Bulk
1N5401-T	DO-201AD	1.2K/Tape & Reel, 13 inch
1N5402-B	DO-201AD	500 Bulk
1N5402-T	DO-201AD	1.2K/Tape & Reel, 13 inch
1N5404-B	DO-201AD	500 Bulk
1N5404-T	DO-201AD	1.2K/Tape & Reel, 13 inch
1N5406-B	DO-201AD	500 Bulk
1N5406-T	DO-201AD	1.2K/Tape & Reel, 13 inch
1N5407-B	DO-201AD	500 Bulk
1N5407-T	DO-201AD	1.2K/Tape & Reel, 13 inch
1N5408-B	DO-201AD	500 Bulk
1N5408-T	DO-201AD	1.2K/Tape & Reel, 13 inch

Notes: 4. For Packaging Details, go to our website at <http://www.diodes.com/datasheets/ap02008.pdf>.

**IMPORTANT NOTICE**

Diodes Incorporated and its subsidiaries reserve the right to make modifications, enhancements, improvements, corrections or other changes without further notice to any product herein. Diodes Incorporated does not assume any liability arising out of the application or use of any product described herein; neither does it convey any license under its patent rights, nor the rights of others. The user of products in such applications shall assume all risks of such use and will agree to hold Diodes Incorporated and all the companies whose products are represented on our website, harmless against all damages.

**LIFE SUPPORT**

Diodes Incorporated products are not authorized for use as critical components in life support devices or systems without the expressed written approval of the President of Diodes Incorporated.



## **B8: ESQUEMÁTICO Y PISTAS PLACA DE ACTUADORES**

