



FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS

DISEÑO Y CONSTRUCCIÓN DE UN DISPOSITIVO LIMITADOR DEL USO
DE LA BOCINA

AUTOR

Patricio Andrés Maldonado Vera

AÑO

2018



FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS

**DISEÑO Y CONSTRUCCIÓN DE UN DISPOSITIVO LIMITADOR DEL USO DE
LA BOCINA**

Trabajo de Titulación presentado en conformidad con los requisitos
establecidos para optar por el título de Ingeniero en Sonido y Acústica

Profesor guía
MSc. Héctor Merino Navarro

Autor
Patricio Andrés Maldonado Vera

Año
2018

DECLARACIÓN DEL PROFESOR GUÍA

Declaro haber dirigido el trabajo, Diseño y construcción de un dispositivo limitador del uso de la bocina, a través de reuniones periódicas con el estudiante Patricio Andrés Maldonado Vera, en el semestre 2018-1, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.

Héctor Merino Navarro
Máster en Posproducción Digital
C.I: 1756785562

DECLARACIÓN DEL PROFESOR CORRECTOR

Declaro haber revisado este trabajo, Diseño y construcción de un dispositivo limitador del uso de la bocina, del estudiante Patricio Andrés Maldonado Vera, en el semestre 2018-1, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación

Luis Alberto Bravo Moncayo
Doctor en Ingeniería Acústica
C.I: 1711710606

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.

Patricio Andrés Maldonado Vera

CI.: 1313123562

AGRADECIMIENTOS

A mi madre, que cree en mí aún en los momentos más críticos, su apoyo y ejemplo son fundamentales para cada emprendimiento en cualquier ámbito de mi vida. Gracias a todas las personas que de algún modo me apoyaron y a las cuales no podía defraudar. A Héctor Merino y Luis Bravo por enaltecer este trabajo. Gracias a la vida por el sonido y el abecedario.

DEDICATORIA

Dedico esta obra a quienes creen que cambiar el mundo está en sus manos, en las pequeñas o grandes acciones que se hacen a favor de la sociedad con los recursos de los que disponen. La dedico también a quienes no encontraron con quién apostar a que no lo lograría. Siempre es bueno tener bocas que callar. Los quiero.

RESUMEN

Los vehículos de transporte han sido dotados de avisadores acústicos desde su origen. La bocina está ahí con la finalidad de permitir al conductor dar un aviso sonoro de peligro a los demás usuarios de la vía. Pero para el caso del transporte terrestre, los bocinazos son una forma común de expresar ira.

En este trabajo se desarrolló un Dispositivo Limitador del Uso de la Bocina (DLUB), que se perfila como una herramienta para regular con eficacia esta mala conducta. Para lograrlo se escribió un programa en Arduino, utilizando *hardware* y *software* de código abierto. De este modo se pudo representar electrónicamente el funcionamiento simplificado de un vehículo circulando a una determinada velocidad, en base a la cual el programa limita la duración de cada bocinazo, considerando una proporción directa entre la rapidez del automotor y el riesgo que podría representar para las personas en su cercanía. Además, el DLUB establece un cupo máximo de tiempo acumulativo para pitar; si un vehículo llegara a superar este cupo, el programa registra una infracción.

Como resultado se obtuvo un aparato que reduce desde su raíz este tipo de contaminación acústica, pues se postula como un método de control de ruido en la fuente, lo que le depara bajo costo social y alta eficiencia. Las pruebas realizadas evidencian que el DLUB permite usar la bocina como una advertencia racional en función del peligro que el vehículo representa, sin embargo, ya que el conductor es consciente de que su bocina está siendo monitoreada y que si abusa de ella será sancionado, evitará descargar su ira con el avisador acústico.

ABSTRACT

Transport vehicles have been provided with acoustic warnings from their beginning. The horn is there in order to allow the driver to give a warning sound of danger to other road users. But in the case of land transport, honking is a common way for expressing anger.

In this work, it has been developed a Limiting Device for the Usage of the Horn (DLUB by its initials in Spanish), which is presented as a tool to effectively regulate this misconduct. To achieve this, a source code has been programmed in Arduino, using open source hardware and software. In this way, the simplified operation of a vehicle circulating at a certain speed has been electronically represented, on the basis of that speed, the program limits the duration of each horn honk, considering a direct proportion between the speed of the vehicle and the risk it could represent for people around it. In addition, the DLUB establishes a maximum quota of cumulative time to use the horn; if a vehicle exceeds this quota, an infraction will be recorded in the program.

As a result, a device that reduces this type of noise pollution was obtained, since it is postulated as a method of noise control at the source, which has low social cost and high efficiency. The tests made show that the DLUB allows the use of the horn as a rational warning depending on the danger that the vehicle represents, however, since the driver is aware that his horn is being monitored, and that if he abuses it will be sanctioned, he will avoid unloading his anger with the acoustic warning.

ÍNDICE

1. CAPÍTULO I. INTRODUCCIÓN	1
1.1. Antecedentes.....	2
1.2. Planteamiento del problema.....	5
1.3. Alcance.....	6
1.4. Justificación.....	6
1.5. Objetivo general.....	7
1.6. Objetivos específicos.....	7
1.7. Metodología a utilizar.....	7
2. CAPÍTULO II. MARCO TEÓRICO	8
2.1. Conceptos acústicos.....	8
2.1.1. Ruido.....	9
2.1.2. Ruido de tráfico.....	9
2.1.3. La bocina como ruido.....	9
2.1.4. Normativa vigente.....	10
2.2. Breve introducción al ecosistema Arduino.....	12
2.2.1. Estructura básica de una placa Arduino.....	13
2.2.1.1. Pines análogos y digitales.....	14
2.2.2. Entorno de desarrollo integrado.....	15
2.2.3. Estructura básica de un sketch.....	17
2.2.4. Funciones.....	17
2.2.4.1. Configuración.....	18
2.3.3.2. Bucle.....	18
2.3.4. Escritura y lectura analógica y digital.....	18
2.3.4.1. pinMode().....	19
2.3.4.2. digitalWrite().....	19
2.3.4.3. digitalRead().....	19
2.3.4.4. analogWrite().....	20

2.3.4.5. analogRead().	20
2.3.4.6. Cambios de estado.	20
2.3.5. Librerías.	21
3. CAPÍTULO III: DESARROLLO	21
3.1. Concepción inicial.	22
3.2. Diseño del prototipo.	22
3.2.1. Selección de la placa Arduino.	25
3.2.2. Circuito externo al Arduino.	25
3.2.2.1. Implementación de LCD.	26
3.3. Diseño del programa.	27
3.2.1. Diagrama de flujo operativo del programa.	28
3.2.2. Código fuente.	30
3.2.2.1. Inclusión de librerías.	30
3.2.2.2. Definición de variables.	30
3.2.2.3. Configuración de inicio.	31
3.2.2.4. Funciones específicas.	32
3.2.2.5. Bucle.	32
3.3. Del prototipo al dispositivo real.	33
4. CAPÍTULO IV. RESULTADOS	33
4.1. Prueba de funcionamiento.	34
4.1.1. Prueba 1: Se supera el cupo dos veces.	34
4.1.2. Prueba 2: No se supera el cupo.	38
4.1.3. Prueba 3: Se varía la velocidad mientras se pita.	40
5. CONCLUSIONES	41
REFERENCIAS	44
ANEXOS	48

1. CAPÍTULO I. INTRODUCCIÓN

La contaminación acústica, a pesar de no ser acumulable como los otros tipos de contaminación, es capaz de causar serios daños en la salud mental y física del ser humano. Es por esto que se ha propiciado la investigación y búsqueda de soluciones por parte de autoridades gubernamentales, organismos internacionales e instituciones educativas.

Se ha demostrado que el ruido de tráfico es la principal fuente de contaminación acústica en áreas urbanas, por encima incluso de otras fuentes importantes como la construcción. Sus efectos son evidentes en personas expuestas a él, como agentes de tránsito, conductores de transporte público y demás personas que trabajan o habitan en las cercanías de calles y avenidas de gran flujo vehicular.

Los vehículos emiten varios ruidos inherentes a su funcionamiento: el motor, el escape, la rodadura (y rozamiento) de las llantas, la turbulencia que generan al desplazarse, etc. Por otro lado, existe un ruido vehicular que es emitido, por lo general, innecesariamente: la bocina. Y es que el avisador acústico está ahí como instrumento de seguridad, con la misión de advertir de un peligro a otros conductores o transeúntes, sin embargo es usado como medio de reclamo, que llama la atención y molesta a quien se está reclamando y a todas las demás personas alrededor del vehículo contaminante, en detrimento de su tranquilidad y calidad de vida.

Este proyecto desarrolla un método de control de ruido de tráfico vehicular. Se trata de un dispositivo electrónico que impide el uso inadecuado de la bocina de los vehículos. Se ha desarrollado el prototipo de un Dispositivo Limitador del Uso de la Bocina (DLUB) con el objetivo de regular la conducta de quienes usan excesivamente el pito del automotor que conducen. Para ello se investigarán casos de aplicación tecnológica que han ayudado a mitigar el ruido vehicular.

1.1. ANTECEDENTES.

Desde los inicios de la era automovilística se ha tratado de regular la contaminación acústica generada por los vehículos motorizados (Grad et al., 1975). No es para menos, el ruido de tráfico es considerado como el principal problema ambiental en las ciudades. Un estudio en las principales calles de Teherán demostró, basándose en cuestionarios aplicados a familiares, que los policías de tránsito sufren deterioro en su comportamiento e incluso desórdenes psicológicos, debido a este tipo de contaminación. Se comprobó también que los agentes están expuestos a niveles sonoros mucho más altos que los 55 dB considerados como aceptables (Omidvari y Nouri, 2009).

El tráfico es la fuente más extendida de ruido ambiental. Se ha demostrado que la exposición al ruido del tráfico está asociada a una amplia gama de efectos sobre la salud y el bienestar humanos. La Organización Mundial de la Salud (OMS) reconoce el ruido del tráfico como un grave problema de salud pública, gracias a lo cual publicó en 1999 directrices para el ruido comunitario. En tal publicación se afirma que el problema de la contaminación acústica también es grave en las ciudades de los países en desarrollo y se debe principalmente al tráfico (Berglund et al., 1999).

Una investigación realizada en seis localidades de Balasore, India, encuestó a 212 personas respecto a ruido de tráfico y reveló que el 47% identificó el dolor de cabeza como el principal efecto de salud de la contaminación acústica y el 39% de los encuestados visitó a doctores especializados en otorrinolaringología para el tratamiento de problemas de salud. En la investigación participaron también estudiantes, el 96% de ellos informó que su estudio ha sido alguna vez interrumpido por los bocinazos frecuentes de vehículos (Goswami, 2009).

Dado lo anterior, el índice de investigaciones en aras de una reducción de ruido de tráfico es basto. Ésta tiene menor costo social y económico cuando se efectúa en la fuente, además de tener mejor relación efectividad/costo. Por

ejemplo, es preferible aplicar silenciadores a los tubos de escape que levantar barreras acústicas frente a residencias colindantes con calles transitadas. Las medidas de reducción más efectivas y de menor costo en este ámbito han sido las relacionadas a la propulsión y llantas de los vehículos (Den Boer y Schroten, 2007).

Por otra parte, la absorción acústica de las superficies porosas de carretera logra una reducción del ruido de rodamiento de hasta en 6 dB(A). Experimentos realizados en Países Bajos muestran que la radiación del ruido aumentada por la geometría del neumático/carretera debe ser tomada en cuenta al predecir la reducción de ruido por una superficie de la carretera que absorba el sonido, para explicar la magnitud y la dependencia de frecuencia específica de la reducción (Peeters et al., 2010)

Pero los vehículos deben también ser acústicamente perceptibles, su presencia debe ser advertida de manera sonora para efectos de seguridad. Es así que las normativas de tránsito incluyen a la bocina dentro de los elementos mínimos de seguridad (ver Normativa Vigente). Además, dados los bajos niveles sonoros de operación en vehículos híbridos y eléctricos, múltiples investigaciones se han centrado en el diseño de sonidos artificiales para que peatones y ciclistas sean efectivamente advertidos del tránsito de automóviles silenciosos (Misdariis et al., 2013; Yamauchi 2016; Parizet et al., 2013; Young Gwak et al., 2014; Kwon Lee y Uk Han, 2017; Uk Han y Kwon Lee, 2017). Tal es la preocupación, que la Comisión Económica de las Naciones Unidas para Europa (UNECE, por sus siglas en inglés) formó un grupo de trabajo para investigar y establecer los lineamientos que los fabricantes de vehículos deben seguir para implementar Sistemas de Alerta Acústica para Vehículos (AVAS, por sus siglas en inglés) (Fortino et al., 2016).

Con la finalidad de aumentar la eficacia de la bocina vehicular como avisador acústico de peligro, investigadores han llevado a cabo estudios que incluyen la implementación de componentes piezoeléctricos flexurales (Kudo et al, 1988),

doble carcasa (Kang et al., 2015) y sobre la psicoacústica del timbre (Lemaitre et al., 2007).

Un estudio en Nueva York afirma que incluso si todos los vehículos fuesen eléctricos, con emisiones inherentes de ruido nulas, la contaminación acústica por tráfico continuaría debido a la congestión vehicular y los bocinazos que esta trae consigo, ya que el comportamiento humano es el factor clave en esta problemática. Tal investigación analiza a profundidad los inconvenientes causados por bocinas y alarmas anti-robo en vehículos. Argumenta que los bocinazos no resultan una molestia en autopistas sino en centros poblados (p. 405), y que en las metrópolis, la bocina es usada para expresar irritación, y no como aviso de peligro. (p. 406). Este mismo estudio asegura que si los conductores están conscientes de que el uso que se da a la bocina está siendo medido y que si la usan excesivamente serán multados, una reducción en el abuso de ésta será inevitable. (p. 436). El artículo concluye que medir y cobrar a los conductores el uso de su bocina es una estrategia efectiva para reducir ese ruido a un costo mínimo para la sociedad. Añade que los automóviles deberían incluir una especie de odómetro que monitoree y desaliente el abuso de la bocina (p. 444) (Brautigam, 1994).

Existe un estudio enfocado en obtener un nuevo y amigable sonido para la bocina vehicular, basado en las preferencias del conductor a través de una prueba de Puntuación de Opinión Media, (*Mean Opinion Score*, MOS en inglés) aplicada a cinco sonidos diferentes. En este estudio, el puntaje se obtuvo en base a las características de la audición humana, pruebas de ondas cerebrales, índice de estrés y respuestas perceptivas de 100 participantes del experimento. De esto se obtuvo un sonido que puede utilizarse en cualquier automotor disponible en el mercado, para que tanto conductores como peatones puedan vivir más cómodamente sin escuchar el ruido molesto de una bocina en las calles (Jee et al., 2017a).

Recientemente, una breve publicación de los mismos autores, plantea un mecanismo simple para dar ritmo a la bocina, interrumpiendo su fuente de alimentación en intervalos de centésimas de segundo. Se evaluó mediante MOS aplicado a 100 participantes, teniendo en cuenta la percepción de riesgo, tamaño del sonido, disgusto y estrés. Como resultado se obtuvo una reducción promedio de -20 dB en el Nivel de Presión Sonora Equivalente y una mayor rapidez en la percepción del riesgo (Jee et al, 2017b).

1.2. PLANTEAMIENTO DEL PROBLEMA

Los atascos de tránsito, conducta agresiva de terceros al volante o cualquier otro motivo para llamar la atención otros usuarios de las vías, ocasionan que los conductores usen la bocina de su vehículo de manera excesiva. La molestia que genera tal ruido tiene consecuencias y en ocasiones inmediatas, como la distracción de otros usuarios, el estrés e incluso riñas entre conductores.

A los automotores se ha incorporado reglamentariamente varios dispositivos para reducir la contaminación, como catalizadores y silenciadores. El DLUB es un dispositivo reductor de contaminación que no afecta el rendimiento del vehículo.

En Ecuador la problemática está en aumento y debido a ello se han promulgado algunas leyes y normativas para regular su uso, entre las cuales están la Constitución del Ecuador, Ley Orgánica de Transporte Terrestre, Tránsito y Seguridad Vial, Ley de Gestión Ambiental y la Ley de Prevención y Control de la Contaminación Ambiental. Además existen entidades y organismos que regulan el uso de la bocina, entre ellas están, la Agencia Nacional de Tránsito, Ministerio de Ambiente, Secretaría de Ambiente, entre otros.

La ley de tránsito considera al uso excesivo de la bocina como una falta leve de primera clase y se sanciona con la reducción de 1,5 puntos en la licencia y el 5% de un Salario Básico Unificado; sin embargo no hay parámetros para

determinar cuándo este uso es excesivo. En la práctica tales sanciones suelen quedar sin efecto.

1.3. ALCANCE.

La presente investigación implica el diseño y construcción del prototipo de un dispositivo electrónico que permita interrumpir el funcionamiento de la bocina de un automotor, basándose en el tiempo de uso diario de ésta y en la velocidad de desplazamiento de éste. Los parámetros que relacionan el límite de tiempo con la velocidad serán definidos de manera arbitraria, pues este proyecto no busca descubrir cuánto de este ruido es molesto, sino brindar un instrumento para evitar que llegue a serlo.

Para esto se utilizará tecnología y programación basada en código abierto que simplifiquen el diseño y la construcción del dispositivo. La implementación se realizará únicamente a nivel de prototipo, suficiente para demostrar el funcionamiento del DLUB de manera didáctica.

1.4. JUSTIFICACIÓN.

A pesar de las normativas vigentes, en el Ecuador persiste el uso excesivo de la bocina, lo cual ocasiona molestias a la ciudadanía, ya que no existe un aparato que regule el uso de la que bien podría ser la fuente de ruido más molesta en el transporte terrestre.

El DLUB se plantea como una solución de control de ruido en la fuente, lo que le depara alta efectividad y bajo costo social, según antecedentes estudiados. Limita el accionar de la bocina en tiempo y tiene en cuenta velocidad del vehículo, además de sancionar a quienes pitan en exceso. Esto supondría un beneficio para la comunidad, en términos de calidad de vida.

1.5. OBJETIVO GENERAL.

Desarrollar un dispositivo que ayude a evitar el uso excesivo de la bocina de un automóvil.

1.6. OBJETIVOS ESPECÍFICOS.

- Describir las normativas vigentes en torno a la regulación del ruido ocasionado por automotores, su alcance y limitaciones.
- Identificar casos de aplicación tecnológica al control de ruido ocasionado por tráfico.
- Examinar el funcionamiento del ecosistema open-source Arduino, en ámbitos de composición electrónica, programación y capacidades de integración.
- Recopilar e integrar códigos de licencia *Creative Commons*, que en conjunto permitan la interrupción de circuitos mediante la activación momentánea de cronómetros.
- Diseñar un complemento para la interfaz de un microcontrolador con dispositivos que le permitan emular aparatos eléctricos del vehículo, como la bocina.
- Implementar un circuito demostrativo compuesto por el sub-circuito tradicional de bocina interrumpido por el DLUB.

1.7. METODOLOGÍA A UTILIZAR.

El nivel de investigación será explicativa, pues ésta se encarga de buscar el porqué de los hechos mediante el establecimiento de relaciones causa-efecto.

En este proyecto se realizará investigación documental, ya que se buscará información en diversas fuentes, tanto impresas como electrónicas. Como técnica se utilizará el análisis documental, utilizando como instrumento la computadora. También investigación experimental, al diseñar y construir un dispositivo electrónico.

Se incursionará en el uso de tecnologías open-source para la elaboración de un sistema denominado **Dispositivo Limitador del Uso de la Bocina (DLUB)**. Como resultado se aspira lograr un prototipo que confronte efectivamente una conducta con gran potencial nocivo al conducir; una herramienta para que las autoridades ambientales y de tránsito identifiquen y sancionen a quienes contaminan acústicamente los entornos urbanos y rurales.

Para evaluar la efectividad del dispositivo se montará un circuito que emule una bocina automovilística, que se accionará a discreción para verificar las interrupciones en función de la velocidad del vehículo, emulada electrónicamente.

2. CAPÍTULO II. MARCO TEÓRICO

Este proyecto integra distintos ámbitos científicos, pues aplica técnicas y conocimientos de electrónica digital y programación, relacionados con la automatización y control de procesos. La confluencia de estos conocimientos pretende solucionar un inconveniente ampliamente estudiado en acústica, la contaminación por ruido de tráfico.

A continuación algunos conceptos necesarios para continuar con el desarrollo de este proyecto a través de las distintas disciplinas involucradas.

2.1. CONCEPTOS ACÚSTICOS.

La acústica estudia, mediante modelos físicos y matemáticos, el sonido y otras ondas elásticas. En acústica ambiental se estudian las causas, consecuencias y soluciones en cuanto a ruido en cualquier ambiente exterior. Para comprender las regulaciones en cuanto a ruido de tráfico es necesario conocer la unidad de medida de ruido por excelencia, el nivel de presión sonora equivalente (L_{eq}).

2.1.1. Ruido.

La definición de ruido como sensación subjetiva, podría reducirse a un sonido indeseado (Sandoval, 2005). Un sonido inadecuado en un lugar inadecuado, en un momento inadecuado. Lo que es música para unos, es ruido para otros.

La contaminación acústica es la más barata de producir, a la vez que necesita muy poca energía para emitirse, pero su medición y cuantificación es compleja, pues no tiene efecto acumulativo en el medio y no se esparce a través de éste, como la contaminación en el agua, por ejemplo. Añadiendo a esto el hecho de que se percibe solamente a través del sentido de la audición, es comprensible que sus efectos sean subestimados (Sandoval, 2005). Sin embargo, su potencial de riesgo empieza a reconocerse, comparándose con la contaminación atmosférica (Fernández, 2000).

2.1.2. Ruido de tráfico.

A nivel mundial, 300 millones de personas sienten incomodidad acústica, causada en un 80% por los vehículos de motor, esto convierte al tráfico en la principal fuente de contaminación acústica (Sandoval, 2005). Pero el motor no es la única fuente de ruido en un automóvil; el rodamiento de los neumáticos, la turbulencia generada a velocidades considerables, y la bocina son otros componentes importantes de la contaminación auditiva producida en un vehículo.

2.1.3. La bocina como ruido.

El sonido producido por la bocina es generalmente catalogado como ruido, pues ha sido diseñado específicamente para elevar el nivel de estrés en los usuarios de las vías y así advertir de un peligro (Kudo et al., 1988; Kang et al., 2015). Contrastando esto con la definición de ruido previamente presentada, queda claro que la bocina es un ruido. Pero es necesario contar con un avisador acústico acorde al peligro que un vehículo en movimiento representa (Lemaitre et al., 2007).

A nivel legislativo, la bocina es generalmente clasificada dentro de los ruidos innecesarios, es decir, que siendo causados por el hecho o acto no derivado de actividad habitual o transitoria del uso normal y adecuado de elementos, son naturaleza de producción superflua, por tanto pueden ser evitados (Miyara, 1999).

2.1.4. Normativa vigente.

A nivel nacional, existe un marco legal que procura la reducción de contaminación acústica en los vehículos. Empezando por la Ley Orgánica de Transporte Terrestre, Tránsito y Seguridad Vial (LOTTTSV), que en su artículo 211 establece:

Todos los automotores que circulen dentro del territorio ecuatoriano deberán estar provistos de partes, componentes y equipos que aseguren que no rebasen los límites máximos permisibles de emisión de gases y ruidos contaminantes establecidos en el Reglamento (Ley Orgánica de Transporte Terrestre, Tránsito y Seguridad Vial, 2014).

En el artículo 322 del Reglamento a la LOTTTSV se señala algo similar, dejando los límites para el Servicio Ecuatoriano de Normalización (INEN):

Todos los automotores que circulen dentro del territorio ecuatoriano, deberán estar provistos de partes, componentes y equipos que aseguren la reducción de la contaminación acústica sin que rebasen los límites máximos permisibles, establecidos en la normativa y reglamentos INEN (Reglamento a la Ley Orgánica de Transporte Terrestre, Tránsito y Seguridad Vial, 2016).

El INEN por su parte, menciona a la bocina en el Reglamento Técnico Ecuatoriano 034 entre los elementos mínimos de seguridad, necesarios para que un vehículo ingrese al parque automotor nacional.

El avisador acústico (bocina) debe ser el original del vehículo y se prohíbe la modificación, alteración o el cambio o adaptación por otro

avisador acústico que incumpla los requisitos mencionados (Reglamento Técnico Ecuatoriano 034 (4R), 2016).

A nivel local, la Ordenanza Municipal 213 del Medio Ambiente del Distrito Metropolitano de Quito, prohíbe en zonas urbanas el uso de la bocina:

Art. II.36S.6.- Se prohíbe la emisión de ruidos que produzcan en las zonas urbanas los dispositivos sonoros, tales como campanas, bocinas, timbres, silbatos o sirenas, instalados en cualquier vehículo, salvo casos de emergencia con la respectiva autorización de la Secretaría del Ambiente (Ordenanza Metropolitana 0213, 2007).

Es importante señalar que hasta 2009, la Ley de Tránsito sancionaba como contravención leve al uso excesivo de la bocina, sin embargo, los artículos que así lo determinaban fueron derogados en la actual LOTTTSV.

Tales artículos indicaban que:

Art. 139.- Incurren en contravención leve de primera clase y serán sancionados con multa equivalente al cinco por ciento de la remuneración básica unificada del trabajador en general y reducción de 1,5 puntos en su licencia de conducir:

a) El conductor que use inadecuada y reiteradamente la bocina u otros dispositivos sonoros contraviniendo las normas establecidas en el Reglamento de la presente Ley y demás normas aplicables, referente a la emisión de ruidos;

Art. 140.- Incurren en contravención leve de segunda clase y serán sancionados con multa equivalente al diez por ciento de la remuneración básica unificada del trabajador en general y reducción de 3 puntos en su licencia de conducir: (...)

r) El conductor que genere ruido por uso excesivo del pito, escapes, u otros (objetos) sonoros;

Siendo la ordenanza municipal el reglamento de mayor jerarquía, resulta imperante en la ciudad. Según la Agencia Metropolitana de Tránsito (AMT), “esta es la cuarta infracción que más se comete en Quito” (Ortega, 2014), sin embargo, a nivel nacional se registran muy pocas sanciones, según la Agencia Nacional de Tránsito (ANT), citada por la misma fuente.

2.2. BREVE INTRODUCCIÓN AL ECOSISTEMA ARDUINO.

Arduino constituye una plataforma para proyectos multidisciplinarios, tanto a nivel de *software* como de *hardware* basada en electrónica y programación de código abierto. Fue concebido ante la ausencia y necesidad de una opción que permita la realización de proyectos con bajo costo y facilidad de uso. (Tu et al., 2010; Evans et al., 2013). Está compuesto por un microcontrolador, un lenguaje de programación y un entorno de desarrollo integrado, que en su conjunto constituyen una herramienta para simplificar el desarrollo de aplicaciones interactivas (Melgar y Diez, 2012).

La comunidad Arduino es realmente grande, en consecuencia, lo son también las fuentes de información y códigos de libre disposición. Todo esto la convierte en la mejor opción para quienes no tienen mayor experiencia previa en programación (Melgar y Diez, 2012). Desde el sitio web oficial provisto de tutoriales fáciles de seguir, hasta una comunidad de usuarios dispuesta a compartir su conocimiento, sin dejar de lado la amplia bibliografía disponible, le dan a la plataforma la accesibilidad y flexibilidad ideal para artistas, diseñadores y creadores. Por ello existen casos de aplicación de Arduino en los ámbitos menos pensados (Evans, 2011).

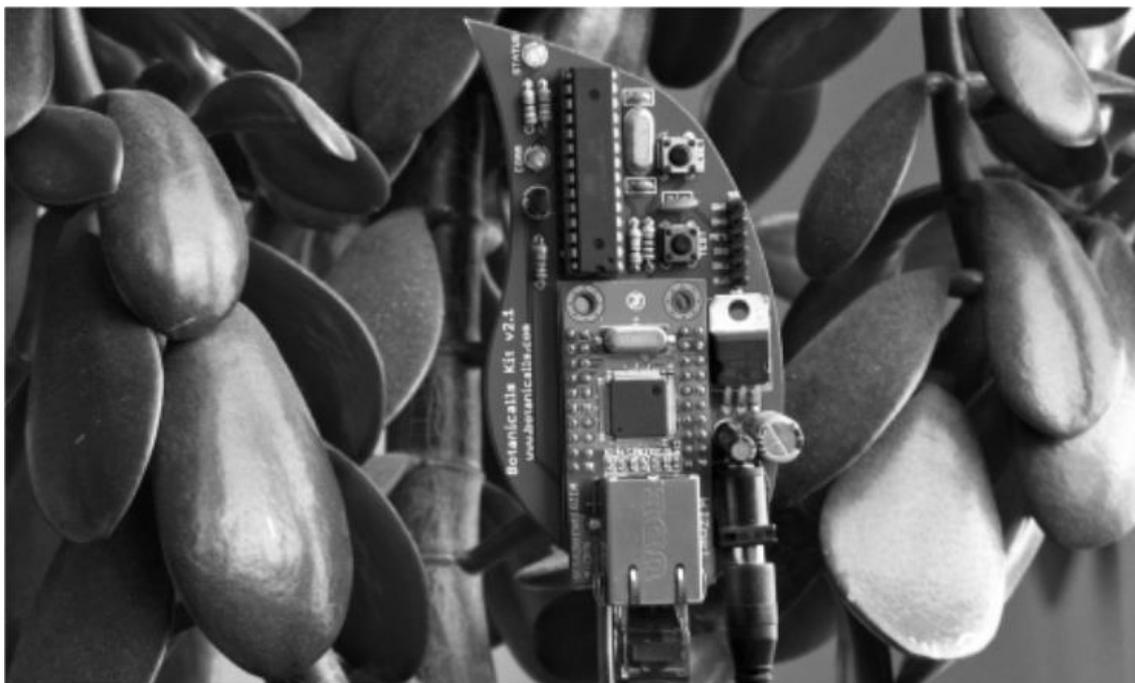


Figura 1. Botanicalls, un dispositivo basado en Arduino que informa cuando una planta necesita ser hidratada.

Tomado de Evans, 2011

2.2.1. Estructura básica de una placa Arduino.

A nivel de hardware, un Arduino consiste en un circuito electrónico que incluye procesador, entradas y salidas digitales y analógicas, y un puerto USB que sirve para comunicación con un computador o para alimentación. El hardware es armado sobre una placa de circuito impreso, hay muchos modelos disponibles en función de la aplicación o funcionalidad y, siendo de código abierto, los esquemas para montarlos están disponibles en línea, pero también se pueden adquirir ya ensamblados (Melgar y Diez, 2012).

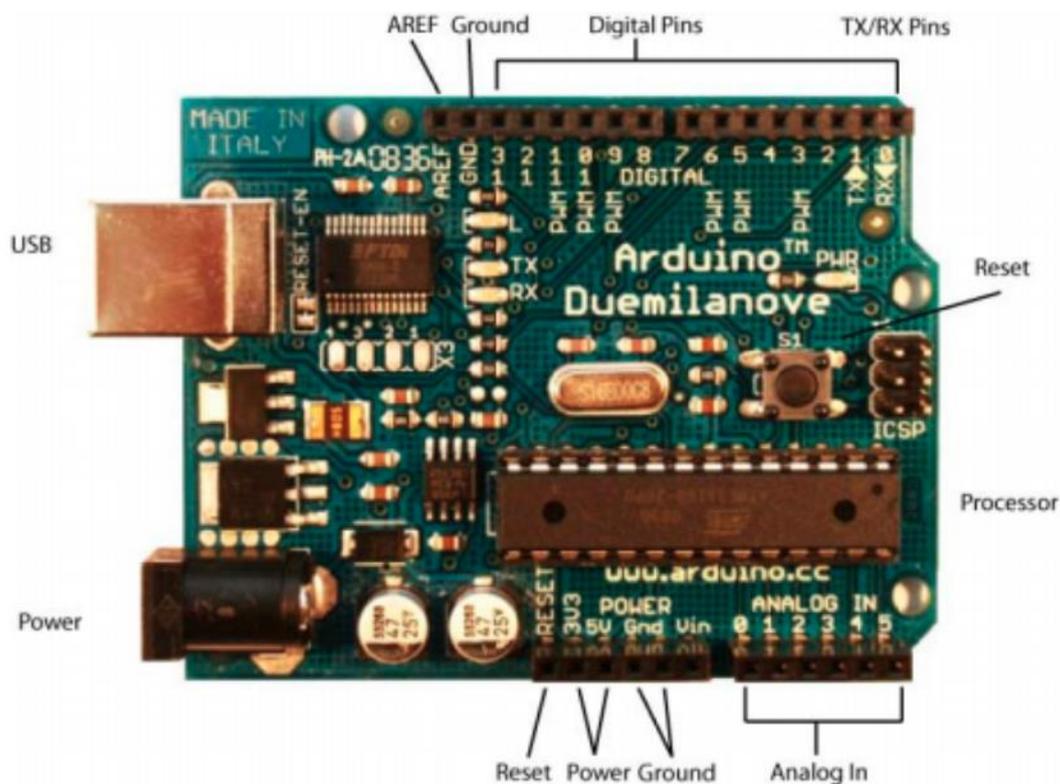


Figura 2. Estructura básica de Arduino Duemilanove.

Tomado de Noble, 2009.

2.2.1.1. Pines analógicos y digitales.

Los pines digitales proporcionan o leen valores digitales de voltaje. Esto significa que solo puede distinguir entre dos posibles valores: *High* y *Low*, siendo el primero 5 voltios (V) y el segundo 0 V. Esto se puede interpretar como encendido / apagado, hay / no hay, listo / no listo; y cualquier otra opción binaria a ser escrita o leída a través de un pin digital (Noble, 2009).

Por su parte, los pines analógicos pueden escribir, cuando se configuran como salidas, 256 valores diferentes de voltaje; mientras que al configurarse como entradas, pueden distinguir 1024 valores distintos. Son utilizados para leer o escribir múltiples valores de una magnitud. Por ejemplo, la distancia de un objeto a un sensor infrarrojo, o la velocidad registrada por los instrumentos de un vehículo (Noble, 2009). El número de pines analógicos y digitales de una placa Arduino varía de modelo a modelo. Los pines analógicos pueden actuar

como digitales, pero los digitales no pueden actuar como análogos. Los pines digitales llevan números como nomenclatura, mientras que los analógicos llevan también un número, pero con una "A" antepuesta.

2.2.2. Entorno de desarrollo integrado.

Por sus siglas en inglés, un IDE (*Integrated Development Environment*), es un *software* diseñado para el desarrollo de más *software* (Melgar et al., 2012). El IDE de Arduino está basado en *Processing*. Este entorno proporciona todo lo necesario para programar un Arduino, incluyendo ejemplos de programas o *sketches* que demuestran cómo conectarlo y comunicarlo con algunos dispositivos comunes como LEDs, LCDs y algunos sensores (Evans et al., 2013).

El IDE de Arduino incluye un editor de código fuente y un compilador que convierte el lenguaje de programación en instrucciones que el microcontrolador pueda ejecutar (Melgar y Diez, 2012).

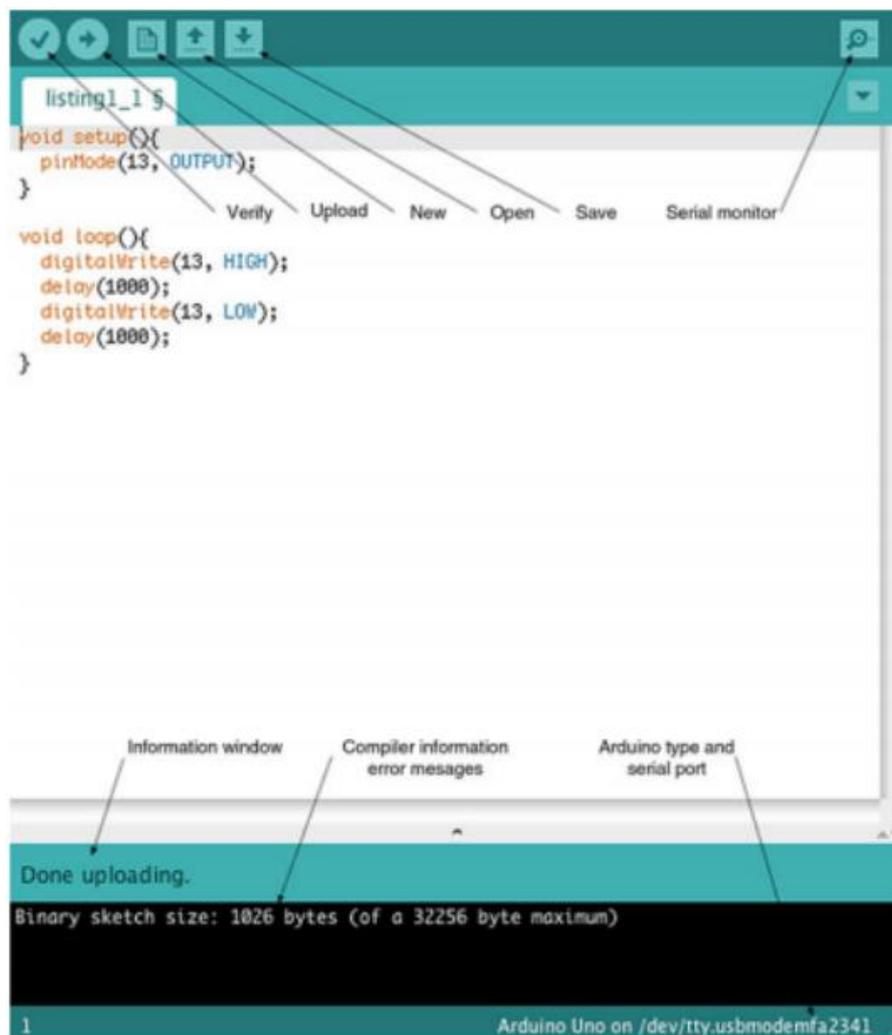


Figura3. IDE Arduino. Se mencionan las partes de la ventana.

Evans et al., 2013.

El botón de verificación *verify* busca errores en el código, que serán reportados en el recuadro negro de la parte inferior de la ventana (*compiler information error messages*: mensajes de error de compilación). El botón *upload* verifica el código y luego lo carga al controlador, asumiendo que está correctamente conectado y configurado. El botón *new* abre un nuevo *sketch*. *Save* guarda el *sketch* y pide nombrarlo en caso de ser necesario. *Serial monitor* abre una nueva ventana con el monitor serie (Evans et al., 2013).

En la parte inferior de la interfaz, además del recuadro de mensajes de error de compilación, existen dos recuadros proveedores de información; el primero

reporta información sobre el estado del programa, mientras que el segundo informa sobre verificación y subida de *sketches* (Melgar y Diez, 2012).

El monitor serie es una ventana parte del IDE que se muestra en la pantalla del ordenador y que permite al Arduino enviar y/o recibir información al computador, en caso de haberse incluido líneas que así lo programen en el código que se encuentra ejecutando (Noble, 2009).

A través del recuadro en la parte superior de la ventana se envía información al dispositivo, mientras que el recuadro más grande debajo de éste, muestra la información que el controlador envía al ordenador (Melgar y Diez, 2012).

Los programas escritos para Arduino se denominan *sketches*, aludiendo al hecho de que se puede escribir un código fuente y fácilmente corregirlo o empezar uno nuevo, como en el proceso artístico de dibujar bocetos (Evans et al., 2013).

2.2.3. Estructura básica de un *sketch*.

El lenguaje de programación Arduino se escribe en base a una librería de funciones llamada *Wiring* que ejecuta comandos en lenguaje C o C++. Por otro lado, la estructura de la programación, es heredada del lenguaje *Processing*. Gracias a ello un programa de Arduino puede reducirse a dos funciones: *setup()* y *loop()*, que serán descritas a continuación (Evans et al., 2013).

2.2.4. Funciones.

Las funciones o sub-rutinas son segmentos de código con inicio y fin, que necesitan ser definidas una sola vez dentro de un programa (Noble, 2009). Para llamar una función, se le antepone el comando *void*, refiriéndose a que no devuelven ningún valor (Smith, 2011). Las funciones son comprendidas a entre llaves { } y normalmente ocupan varias líneas de programa. En Arduino, las funciones más utilizadas son las siguientes:

2.2.4.1. Configuración.

Los sketches de Arduino deben prepararse o configurarse antes de empezar (Evans et al., 2013). *Setup()* es llamada una sola vez al inicio del código y su función es configurar varios aspectos necesarios en la placa Arduino para que el programa pueda correr apropiadamente (Evans, 2011). Este es el lugar para todo lo que necesita ser inicializado, comunicación serial, comunicación con LCD y todo lo que necesita recibir una señal antes de ser usado (Noble, 2009).

2.3.3.2. Bucle.

La función *loop()* o bucle, contiene todo lo que necesita ocurrir de manera repetitiva en el programa; leer valores de un sensor, enviar información a un computador, enviar señal a un pin, o enviar información de depuración (Noble, 2009). Todas las instrucciones dentro de un bucle serán ejecutadas una y otra vez hasta que el programa termine. Esta función resulta ser la esencia de la mayoría de programas en Arduino (Melgar y Diez, 2012). La función *loop* se ejecutará hasta que una condición sea alcanzada o la placa Arduino sea apagada (Evans et al., 2013).

2.3.4. Escritura y lectura analógica y digital.

Como se describió anteriormente, los pines de una placa Arduino pueden configurarse como entradas o salidas digitales, y además como entradas o salidas análogas en el caso de los pines analógicos (Smith, 2011). Para leer o escribir digitalmente en un pin, es necesario dejar saber previamente al microcontrolador, si será una entrada o salida, para lo cual se usará la función *pinMode()*.

2.3.4.1. *pinMode()*.

Un pin digital puede ser usado como entrada (*INPUT*) o salida (*OUTPUT*), lo que le permitirá enviar o recibir valores de un microcontrolador. Esto se realiza normalmente en la sección *setup()*, ya que raras veces un pin alternará entre entrada y salida en el programa (Noble, 2009).

Cuando un pin es asignado como entrada, el microcontrolador le configura un estado de alta impedancia, lo que le permite recibir una señal entrante. Siendo digital, la entrada es sensible solo a dos valores, *HIGH* y *LOW* (Evans, 2011).

Por el contrario, cuando se asigna al pin como salida, se le proporciona baja impedancia, lo que le permite brindar una corriente suficiente para encender un LED mientras provee una pequeña resistencia al resto del circuito (Evans, 2011).

2.3.4.2. *digitalWrite()*.

Esta función tiene como pre-requisito haber declarado el pin en cuestión como salida (*OUTPUT*). Solo entonces se puede usar la escritura digital para definir un estado *HIGH* o *LOW* en un pin específico. En el primero se conduce +5V al pin, mientras que en el segundo, el valor predefinido de cualquier salida, el voltaje se anula conectándolo a tierra (Evans, 2011). Con esto se puede encender o apagar cualquier elemento electrónico que funcione a cinco voltios y que sea compatible con la intensidad de corriente que puede brindar ese modelo específico de Arduino.

2.3.4.3. *digitalRead()*.

Una vez declarado un pin como entrada o *INPUT*, es posible usar la lectura digital para interpretar si es que, por ejemplo, un sensor conectado a dicho pin está encendido o apagado. Es común que esta función sea usada para asignar el valor de una lectura a una variable previamente declarada, (Evans, 2011).

2.3.4.4. *analogWrite()*.

Esta función se puede ejecutar solo en un pin analógico. Permite asignar en dicho pin 256 valores distintos de voltaje (de 0 a 255). Esto hace posible usar un voltaje ajustable para manejar un dispositivo que así lo requiera (Noble, 2009).

2.3.4.5. *analogRead()*.

La lectura analógica brinda una gama de 1024 valores distintos (desde 0 hasta 1023), en representación del voltaje medido en el pin. Los dispositivos conectados al Arduino deben enviar señales entre 0 y 5V; valores fuera del rango no serán leídos y podrían dañar la placa. Esta función se puede ejecutar solamente en pines analógicos y normalmente su lectura se guarda en una variable entera previamente declarada (Noble, 2009).

2.3.4.6. Cambios de estado.

Dados los dos posibles estados en una entrada o salida digital, es posible detectar un cambio de estado, es decir, cuando un pin cambia de encendido a apagado, o viceversa, e incluso contar estos cambios. Para esto, no se necesita saber en qué estado se encuentra un pin, lo que deseamos conocer es, si en la lectura anterior, el estado era distinto al actual (Evans, 2011).

Si al leer una entrada, a la cual se encuentra conectado un botón, la lectura actual es encendido (*HIGH*), pero en la anterior lectura era apagado (*LOW*), entonces el botón acaba de ser presionado. Por el contrario, si la lectura actual es *LOW*, pero la anterior fue *HIGH*, significa que el botón acaba de ser soltado. Realizar esta observación es lo que se llama detección de corte (*edge detection*); ya que lo que realmente se hace es detectar el momento específico o corte en que el estado cambia de una condición a otra (Evans, 2011).

2.3.5. Librerías.

Una librería es una colección de códigos que pueden ser usados por muchos programas. Esto permite llamar funciones para facilitar el cumplimiento de diversas tareas (Smith, 2011). Existen librerías extremadamente útiles que permiten, por ejemplo, trabajar con LEDs, controlar motores, usar pantallas LCD e incluso controlar teléfonos celulares (Noble, 2009). Se las llama librerías en alusión a la analogía de un investigador que usa de referencia otros trabajos previamente realizados para la ejecución del suyo; este investigador tomará de la librería las partes que necesite en su proyecto (Evans et al, 2013).

El IDE incluye varias librerías que el equipo de desarrollo de Arduino considera necesarias para muchos proyectos. Para usar una librería en un código, es necesario “incluirla” en las primeras líneas del mismo, ya que no se incluyen por defecto ya que sería un desperdicio de recursos. Al llamar una librería mediante `#include <NombreDeLibreria.h>`, le pedimos al IDE incluir implícitamente un conjunto de subrutinas en el código, cuya programación simplificará tareas específicas sin engrosar las líneas del *sketch*. Las librerías contribuidas por la comunidad que no se distribuyen con el IDE necesitan primero ser descargadas al computador (Evans et al, 2013).

3. CAPÍTULO III: DESARROLLO

El Dispositivo Limitador del Uso de la Bocina fue concebido por el autor de esta tesis como una propuesta contundente, creativa y viable para la disminución de contaminación sonora causada por bocinas de automotores.

El proceso creativo de idealización del dispositivo se dio sin mayor investigación de antecedentes, quizás el anhelo de regular una conducta altamente dañina hizo que la sencillez del aparato se muestre como solución evidente. En concepto, el prototipo pasó por una versión preliminar, producto neto de la inventiva del autor, idea que al plantearse como proyecto de tesis

evolucionó para lograr un método de control de ruido que permita a la bocina realizar una advertencia acústica acorde al peligro que el vehículo representa.

3.1. CONCEPCIÓN INICIAL.

Inicialmente se planteó como un dispositivo que silenciara el claxon una vez superado un cupo diario de tiempo acumulativo para pitar; dicho cupo era de siete segundos. Previo a la presentación de la idea, se tomó en consideración una evidente desventaja para el caso de los transportistas, que circulan más horas al día, por lo que se añadió la función de recuperación de un segundo en el cupo usado por cada 10 km de recorrido del vehículo. El pito quedaba deshabilitado hasta el día siguiente y el DLUB registraba una multa por cada día en que se hubiera agotado el cupo.

En el planteamiento de esta idea ante el tutor de tesis, éste hizo observaciones respecto al problema que implicaría dejar un vehículo sin la posibilidad de realizar una advertencia sonora, recomendando luego que, en lugar de enmudecer totalmente la bocina, el DLUB limite cada uno de los pitidos en función de la velocidad a la que se desplace el vehículo, puesto que dicha magnitud guarda relación directa al peligro que un vehículo podría representar a otros usuarios de las vías. Fue así que se definió el dispositivo respecto al cual está elaborado este documento.

3.2. DISEÑO DEL PROTOTIPO.

El DLUB debe ayudar a disminuir la contaminación acústica ocasionada por el abuso de la bocina de los automotores. Pero este aparato no se puede concebir como un mutilador de bocinas, pues sin importar cuán molesto sea el comportamiento de un conductor, puede necesitar advertir de un peligro a otros usuarios de la vía. Deberá permitir que se use la bocina como una razonable advertencia sonora, cuya molestia sea equiparable al riesgo que se anuncia al pitar.

El principal antecedente de este dispositivo proponía inspecciones anuales del uso de la bocina, dejando un margen razonable para casos de emergencia, que una vez superado, represente una sanción económica. Esta modalidad traería inconvenientes, por ejemplo, en el caso de los vehículos de alquiler, que anualmente tienen cientos de usuarios, por lo que resultaría imposible determinar cuáles de ellos usaron razonablemente la bocina y cuáles de ellos ameritan ser sancionados. (Brautigam, 1994).

El DLUB, en cambio, propone un cupo diario que al superarse guarda un registro con información oportuna que ayudaría a las compañías de alquiler a determinar qué usuario incurrió en uso excesivo de la bocina, para así imputarle la sanción económica que el vehículo gravó.

La siguiente tabla muestra las variables determinantes en el funcionamiento del dispositivo, así como los elementos lógicos y físicos (de ser el caso) que las representan:

Tabla 1.

Principales variables y su representación en el dispositivo.

Variable	Elemento físico	Variable Arduino
Velocidad	Potenciómetro	speedSensorValue
Velocidad máxima		maxSpeed
Duración bocinazo	Altavoz Piezoeléctrico	totalSecClaxonOn
Tiempo total bocinazos al día		totalTimeAtDay
Segundos por Kilómetros		secPerKilometerToLimit
Cupo diario de tiempo para pitar		maxTimeClaxonOnAtDay

Este producto tiene como clientes potenciales a entidades gubernamentales como las de control de ambiente o tránsito, a quienes debe proporcionar

información sobre las infracciones en que el conductor haya incurrido. Por otro lado, están los conductores, a quienes se deberá mantener al tanto del registro de su conducta con la bocina. En detalle, las funciones que debe cumplir el dispositivo son:

- Limitar la duración de cada *accionar* de la bocina en función de la velocidad del vehículo. Es decir, establecer un tiempo máximo luego del cual se desconectará temporalmente la bocina, dependiendo de qué tan llamativa tenga que ser la advertencia sonora. Un vehículo andando a 100 Km/h necesita una advertencia más llamativa que otro circulando lentamente en un embotellamiento de tráfico y que, más que precautelar, quiere reclamar.
- Llevar un registro del uso de la bocina sumando los tiempos de cada pitido en un contador. Este contador deberá compararse con un cupo diario y mostrar tal comparación al conductor del vehículo para mantenerlo informado y que este pueda evitar ser sancionado.
- Guardar un registro cada vez que el conductor haya superado el cupo establecido. Este registro debe ser accesible para la autoridad competente y servir como evidencia para una sanción.
- Debe tener autonomía energética que le permita funcionar independientemente del funcionamiento del vehículo; sin perder el registro de las infracciones, ni la hora y fecha. Idealmente deberá también tener seguridad ante intentos de alteraciones por parte de usuarios o técnicos no autorizados.

El producto está conformado por un circuito electrónico manejado por un microcontrolador Arduino, y la programación que éste ejecuta para ejercer regulación sobre el uso de la bocina.

Como cerebro del DLUB se tiene a la placa Arduino, que se comunica con un circuito ejecutor. Tal circuito emula la bocina y el velocímetro del automóvil, cuenta además con una pantalla LCD, mediante la que proporciona

información al usuario y cuenta con dos pulsadores, uno para accionar la bocina y otro para imprimir en el ordenador el registro de infracciones.

3.2.1. Selección de la placa Arduino.

El modelo más usado por la comunidad de usuarios Arduino, el más accesible y el más replicado, el UNO presenta características suficientes para implementar el DLUB, entre ellas resalta su procesador ATmega328, con capacidad de almacenamiento relativamente amplia, además de contar con 14 pines de entrada/salida digital y seis entradas analógicas, siendo estos los números más cercanos a la cantidad de entradas y salidas que requiere el prototipo.

3.2.2. Circuito externo al Arduino.

El circuito externo debe, en primer lugar, entregar al Arduino un voltaje que el microcontrolador interprete como la velocidad del vehículo. Tal voltaje se podrá variar mediante un potenciómetro. Además, este circuito integra un altavoz piezoeléctrico, que hace las veces del claxon, y que es accionado por un pulsador a través de un transistor.

El circuito debe encender una advertencia si el cupo de tiempo para pitar está por agotarse, y mostrar otra señal en caso de que se haya acabado ya el tiempo límite diario para pitar; dos LEDs de color amarillo y rojo respectivamente se encargarán de ejecutar tal función.

Además, este circuito da la orden al Arduino para que éste envíe a un computador conectado por USB los registros de las faltas cometidas, para ello se incluye un pulsador adicional. Los elementos mencionados junto con sus respectivas resistencias para controlar voltajes, conforman el circuito externo al arduino.

3.3. DISEÑO DEL PROGRAMA.

El programa cargado en el microcontrolador se encarga de que éste use sus recursos y los de la placa Arduino para que los demás elementos electrónicos interactúen con ella y desempeñen en conjunto las funciones para las que fue concebido el prototipo.

El Arduino debe conocer el estado de los dos pulsadores del circuito ejecutor, así como la velocidad a la que se mueve el vehículo, representada por una señal analógica entregada por el potenciómetro. A esto le agrega información de hora y fecha, generada internamente mediante librerías y funciones específicas. Con todos estos datos el programa genera información en base a la cual deberá accionar, cronometrar y apagar la bocina; almacenar en variables un registro del uso total de la bocina en el día; imprimir en la pantalla LCD datos útiles para el usuario; encender o apagar luces de advertencia; almacenar en la memoria del microcontrolador la fecha, hora y velocidad, en caso de que se supere el cupo diario, registrando cada infracción; y enviando al monitor serie el registro del total de multas hasta la fecha de consulta.

3.2.1. Diagrama de flujo operativo del programa.

Se representa a continuación, de manera simplificada, el funcionamiento del DLUB. En el primer diagrama se detallan los datos que el Arduino lee, tanto analógicos como digitales. En el segundo se explica el proceso de lectura del *switch* de la bocina.

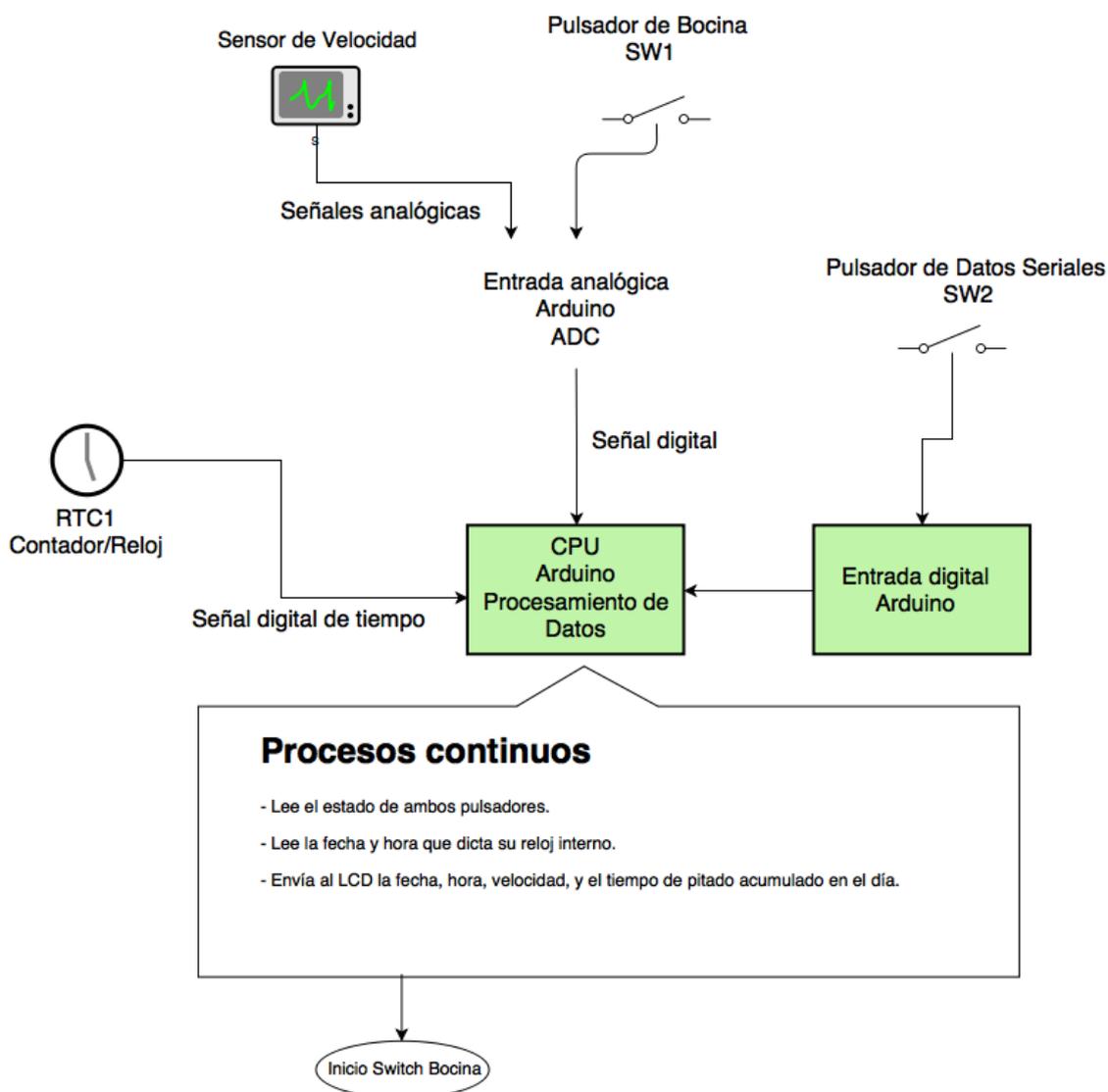


Figura 5. Señales que ingresan al microcontrolador. RTC1 representa al reloj de tiempo real generado internamente por el procesador a través de la librería *TimerOne*.

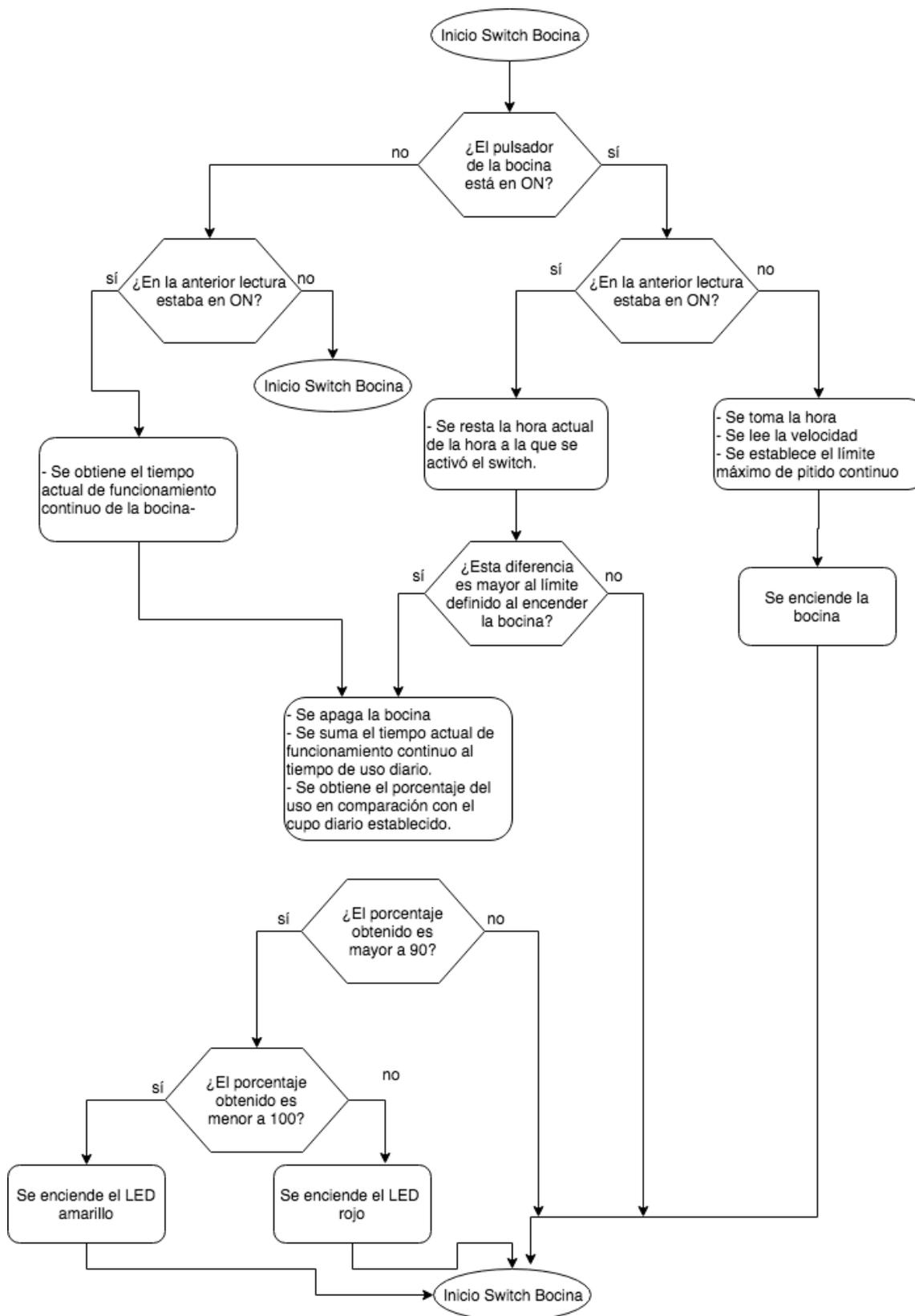


Figura 6. Diagrama de flujo del proceso ejecutado al accionar el pulsador de la bocina.

3.2.2. Código fuente.

A continuación se describe cada segmento del código fuente, mismo que se encuentra como primer apartado de la sección Anexos.

3.2.2.1. Inclusión de librerías.

Para simplificar la programación se aprovechó la naturaleza *open source* de Arduino para incluir librerías elaboradas previamente por otros usuarios de la comunidad. Las tres librerías a continuación permiten la utilización de los recursos del microcontrolador para ejecutar funciones cada cierto tiempo; correr un reloj de tiempo real y su respectivo calendario; y comunicación con el módulo LCD, respectivamente:

3.2.2.2. Definición de variables.

Con el propósito de que el código pueda ser fácilmente interpretado por la comunidad Arduino, se ha escrito los nombres de variables y funciones en inglés. A continuación su desglose.

Se definen los pines digitales 5, 6 y 2 para el LED anaranjado de advertencia, el rojo de infracción y la bocina, “alarmPinOrange”, “alarmPinRed” y “claxonPin”, respectivamente. Luego se definen los valores constantes y arbitrarios de conversión de kilómetros por hora a segundos para limitar el accionar de la bocina “secPerKilometerToLimit”; y el cupo máximo de tiempo diario para pitar “maxTimeClaxonOnAtDay”, valores flotantes para poder usar decimales en las operaciones aritméticas a realizarse. Similar definición se practica para la velocidad máxima del vehículo “maxSpeed”.

Se define como variable a la máxima lectura analógica de velocidad “maxSpeedADCValue”. Tal velocidad será leída a través del pin analógico A0, a partir de ahora “speedSensorPin”, mientras que al pulsador de la bocina le corresponderá la A1 “claxonSensorPin”. Para la lectura de estas entradas se

definen las variables “speedSensorValue” y “claxonSensorValue” respectivamente.

Se definen variables para las unidades de tiempo, desde los milisegundos hasta el año. Las variables “pos” y “pos2” se usarán para posicionar información en el módulo LCD. Se define también el arreglo “memory” con 31 registros de 9 datos, más la variable “memoryCont” para contar los registros ingresados.

Se define una variable bandera “flag” booleana, que será usada más adelante para conocer el estado en la anterior lectura del pulsador de la bocina. Dos variables flotantes se usarán para contar los segundos de funcionamiento continuo del claxon: “totalSecClaxonOn” será usada para el caso en que se suelte el pulsador antes de alcanzar el límite basado en velocidad, mientras que “totalSecClaxonOn2” en el caso de que se alcance el límite sin haber apagado el pulsador. El mismo tipo de variable será usado para almacenar el tiempo total de uso diario “totalTimeAtDay”, el porcentaje que éste representa respecto al cupo diario establecido “percentClaxonOnAtDay” y el límite de uso continuo, calculado a partir de la velocidad, será “speedLimitInSec”.

3.2.2.3. Configuración de inicio.

En esta sección se define la variable “maxSpeedADCValue” mediante un artificio que la hará compatible con la conversión análoga/digital. Se inicializa la función de fecha y hora con las variables enteras “y”, “m”, “d”, “h”, “mi” y “se”, que representarán operativamente al año, mes, día, hora, minuto y segundo. Se inicializa el LCD también, indicando que tendrá 16 columnas y dos filas. Luego se definen como salidas a los pines 5, 6 y 2, ahora llamados “alarmPinOrange”, “alarmPinRed” y “claxonPin”; mientras que el 4 “buttonPin” es definido como entrada.

Se inicializa ahora la librería nativa “Timer1” con periodos de un segundo, entre los cuales ejecutará la actualización del reloj programado más adelante. Se inicializa también la comunicación serial a 9600 Hz. Aquí se programa a la

librería “MsTimer2” para ejecutar la función “getADCValues”, detallada más adelante.

3.2.2.4. Funciones específicas.

La primera función específica del código es “ISR_Clock”, que a su vez ejecuta tres funciones: Actualizar el reloj, mostrar el tiempo los datos actualizados sobre el uso de la bocina en el LCD.

La siguiente es “getADCValues, cuyo papel es leer los pines analógicos de la velocidad, del pulsador de la bocina y realizar la conversión de variable entero a flotante para speedSensorValue.

“setClaxonTimeInfoLCD” Se encarga de ubicar en la celda correspondiente del LCD la información, además de encender el LED amarillo en caso de superarse el 90% del cupo diario, o el rojo en caso de superarse el 100% del cupo. En caso de superarse el cupo, llama a “setDataMatrix”, que se encarga de registrar la fecha, hora, tiempo de uso acumulado del día y su porcentaje, información importante para validar la infracción.

Un reloj con milisegundos, segundos, minutos, horas, fecha, mes y año se ejecuta a través de las funciones “update_msec” y “update_clock”.

3.2.2.5. Bucle.

En esta sección, previamente explicada mediante diagrama de flujo, se empieza preguntando si el pulsador se acaba de encender, para lo cual se vale de la variable booleana *flag* (bandera). Si se acaba de encender el pulsador, se enciende la bandera, se indica en el monitor serial que la bocina acaba de encenderse, se registra la hora de encendido, se registra la velocidad y en base a ésta se establece el límite de pitido continuo.

Se pregunta si es que el claxon está encendido y ya lo estaba en la anterior lectura. De ser así, se registra la hora actual en una nueva variable y se resta de la hora a la que se encendió. Si la diferencia es mayor al límite establecido al encenderse la bocina, se la desconecta.

Se pregunta si se acaba de apagar el pulsador de la bocina, en cuyo caso realizará lo siguiente:

- Se apaga la bocina.
- Se apaga la bandera.
- Se indica que se ha apagado el claxon en el monitor serie.
- Se obtiene el tiempo actual de funcionamiento continuo de la bocina.
- Se suma el tiempo actual de funcionamiento continuo al tiempo de uso diario.

3.3. DEL PROTOTIPO AL DISPOSITIVO REAL.

Para la implementación del dispositivo final, que pueda ser incorporado a un vehículo para regular el uso de una bocina real, bastaría con reemplazar a la bocina con un relevador o relé. Esto permitiría que el circuito del dispositivo, que maneja voltajes y amperajes de escala digital, controle la potencia superior de una bocina de automóvil. El dispositivo debería encerrarse en una caja que lo proteja de la humedad, calor y manipulación por parte de cualquier persona que no sea quien toma nota del registro de infracciones de la bocina en el centro de revisión vehicular.

4. CAPÍTULO IV. RESULTADOS

Una vez ensamblado y programado el DLUB, se procedió a realizar pruebas de su funcionamiento, con el fin de verificar que la programación logre los objetivos planteados.

4.1. Prueba de funcionamiento.

Para poner en funcionamiento el dispositivo, se lo conectará mediante el cable USB a un ordenador con el IDE Arduino instalado, obteniendo así un monitor serie y fuente de alimentación a la vez.

Inicialmente, el aparato muestra en pantalla información sobre la hora y fecha, así como la velocidad del vehículo y el uso acumulado de la bocina en el día y el porcentaje que éste representa al compararse con el cupo diario establecido. Las pruebas se ejecutarán pitando siempre hasta que se desconecte la bocina, es decir, hasta sumar, en cada ocasión, una centésima de segundo por cada kilómetro por hora de velocidad del automóvil. Se experimentará a distintas velocidades arbitrarias con el propósito de demostrar los distintos estados de advertencia e información proporcionados por el dispositivo.



Figura 7. Información inicial en pantalla LCD.

4.1.1. Prueba 1: Se supera el cupo dos veces.

Se accionó la bocina una vez y, ya que la velocidad era de 60 Km/h se suma 0.6 s al tiempo acumulado. Siendo este el primer pitido registrado, el total de uso acumulado resulta 0.6 s, equivalente al 12% del cupo diario. A continuación se cambió la velocidad mediante el potenciómetro a 90 Km/h, se accionó la bocina una vez, por lo que se sumó 0.9 s. al uso acumulado, resultando 1.5 s. Luego se estableció una velocidad de 100 Km/h, por lo que luego de pitar una vez más, el uso acumulado es 2.5 s. Se pitó dos veces más a esta velocidad, consiguiendo un uso acumulado de 4.5 s., que corresponde al

90%, el LED amarillo se enciende, advirtiendo que si se sigue usando la bocina podría registrarse una multa.

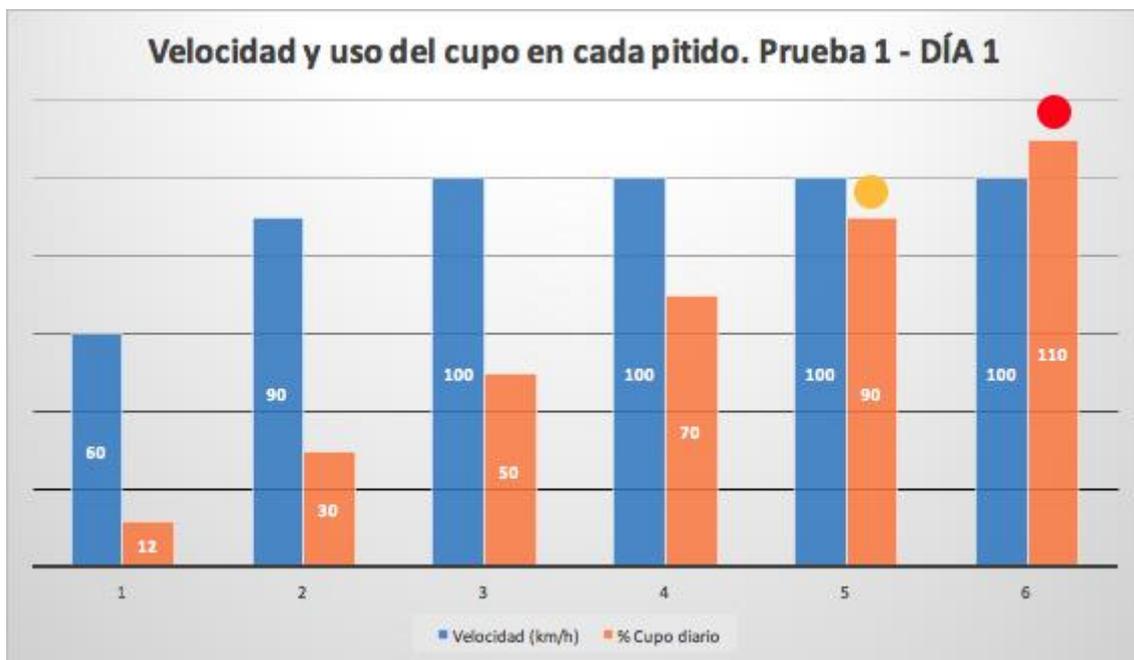


Figura 8. Resultados del primer día para la prueba N° 1. En azul se representa la velocidad al iniciar cada pitido, en naranja el porcentaje de uso acumulado del cupo diario al terminar cada uno de los seis pitidos.



Figura 9. Se alcanza el 90% del cupo. Se enciende el LED amarillo.

Se pitó una vez más a la velocidad de 100 Km/h, con lo que el uso acumulado llegó a 5.5 s., y ya que es superior al límite establecido de cinco segundos, se enciende el LED rojo, indicando que se ha registrado una multa.



Figura 10. Se continúa pitando hasta sobrepasar el cupo. Se enciende el LED rojo.

Transcurren unos minutos y, al ser las cero horas del día siguiente, se muestra un uso acumulado de 0 s, debido al reinicio del contador, como consecuencia del cambio de fecha.

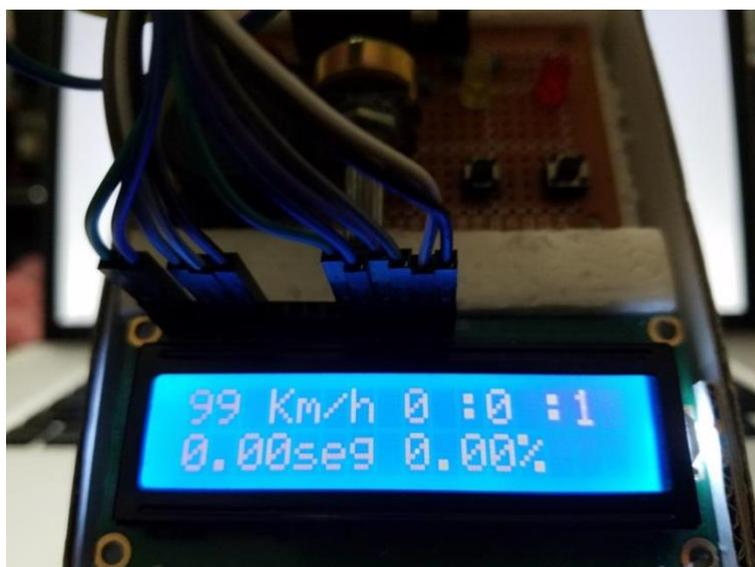


Figura 11. Al cambiar de fecha, el uso acumulado se muestra en cero.

Una vez más, se acciona la bocina a velocidad de 180 Km/h tres veces, superando el cupo, por lo que el LED rojo se vuelve a encender.



Figura 12. Resultados del segundo día para la prueba N° 1. Se supera el cupo luego de pitar tres veces.

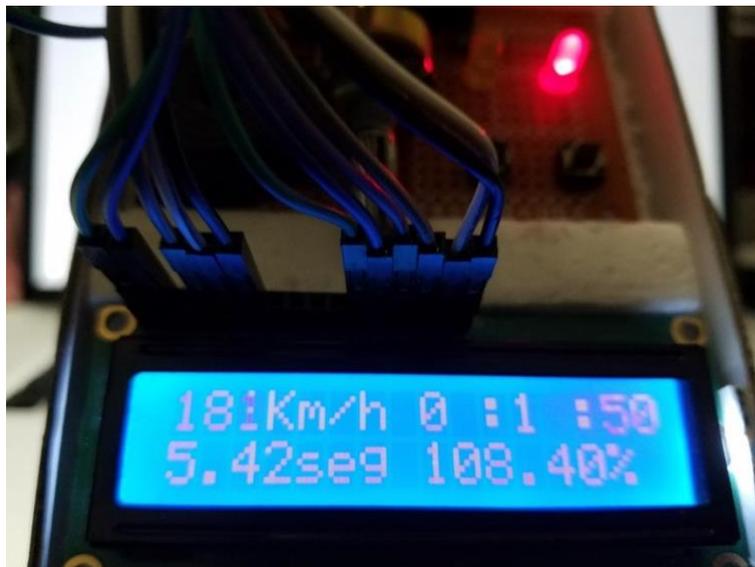


Figura 13. Se pita nuevamente hasta superar el cupo diario. Se enciende el LED rojo.

Se presiona el microswitch S2, como resultado, en el monitor serial del ordenador se imprime el mensaje sobre las dos infracciones registradas hasta la fecha.

```

/dev/cu.usbmodemFD121 (Arduino/Genuino Uno)
Enviar
Claxon On! Claxon Off!
*****Registro de Infracciones*****
Date: 5/12/16 Time: 23:56:35 Total Time: 5 seconds Percent: 109 % Speed: 100 Km/h
Date: 6/12/16 Time: 0:0:48 Total Time: 5 seconds Percent: 107 % Speed: 179 Km/h
*****
Autoscroll Nueva línea 9600 baudio

```

Figura 14. Se imprime en el monitor serie las infracciones registradas.

4.1.2. Prueba 2: No se supera el cupo.

Se accionó la bocina una vez a 10 Km/h, velocidad típica de situaciones de congestión vehicular. Siendo este el primer pitido registrado, el total de uso acumulado resulta 0.1 s., equivalente al 2% del cupo diario. Se modificó la velocidad y ahora es 57 Km/h, apenas por encima del límite de velocidad urbana en Ecuador, se accionó la bocina una vez, por lo que el uso acumulado es ahora 0.67 s, equivalente al 13.4%. La velocidad es ahora 20 Km/h, por lo que luego de pitar cinco veces más el uso acumulado es 1.87 s. o el 37.4%.

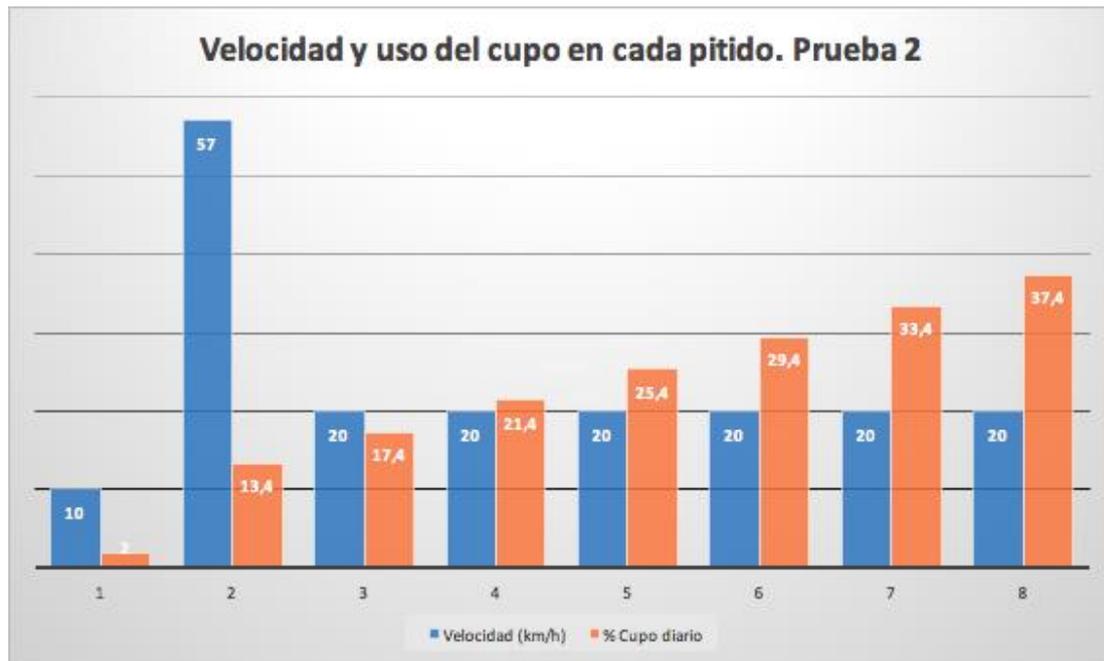


Figura 15. Resultados para la prueba N° 2. Se pita en 8 ocasiones, sin embargo el cupo no llega a ser superado.



Figura 16. En la segunda prueba no se supera el cupo.

Se presionó el microswitch S2, en el monitor serial del ordenador se imprimen las infracciones registradas hasta la fecha. El registro aparece en blanco.

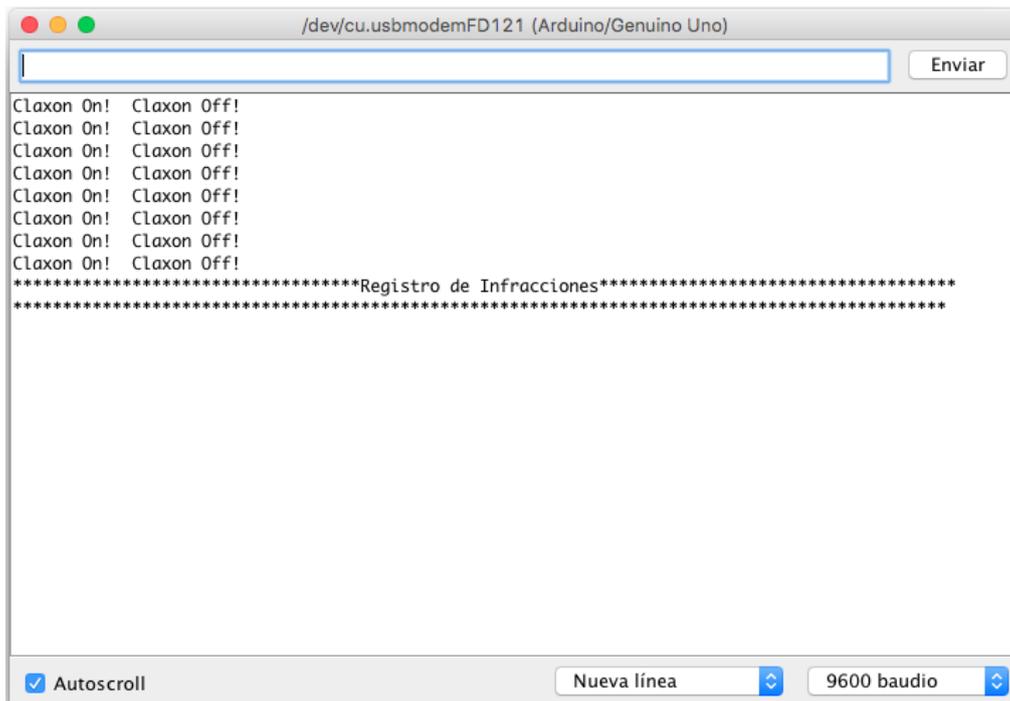


Figura 17. Impresión en el monitor serie de las infracciones, sin registros.

4.1.3. Prueba 3: Se varía la velocidad mientras se pita.

En esta última prueba, se emula la siguiente situación particular: el vehículo se encuentra circulando al límite de velocidad en carreteras, un peligro se presenta en su trayecto, por lo que la reacción de quien lo conduce es frenar mientras pita reiteradamente. Para facilidad de comprensión y cálculo, se asume en este ejercicio que el vehículo frena linealmente, es decir, que su tasa de frenado es constante. Se asume también un intervalo de medio segundo (0.5 s) entre pitido.

La simulación inicia fijando la velocidad en 100 Km/h, luego se acciona la bocina a la vez que se reduce la velocidad hacia cero. Como se esperaba, el bocinazo es interrumpido luego de un segundo. Luego se suelta el pulsador de la bocina para volver a pulsarlo, siendo la velocidad cercana a cero, el DLUB desactiva la bocina luego de una centésima de segundo.

5. CONCLUSIONES

Este trabajo desarrolló un producto de gran utilidad como método de control de ruido en la fuente para tráfico. No se halló evidencia de desarrollos tecnológicos que le antecedan en su objetivo, por lo que que marca un importante punto de partida hacia el monitoreo y regulación electrónico del uso de la bocina de los automóviles. Además, entre los antecedentes citados, constan investigaciones que justifican a plenitud la presente investigación; desde postular la necesidad de un odómetro para avisadores acústicos, hasta la verificación de la reducción de molestia y aumento de efectividad en comunicación de riesgos al interrumpir el sonido. Se puede suponer, por lo tanto, que esta investigación contribuirá significativamente a la comunidad.

Se diseñó y ensambló una interfaz electrónica que permite controlar dispositivos. Sin embargo se optó por no incorporar directamente elementos de control para la bocina, dadas las dificultades que éstos implican en términos didácticos. En su lugar, se incorporó un parlante piezoeléctrico como emulador de la bocina.

El programa tiene dos variables fácilmente modificables, en caso de que se quiera cambiar la relación entre la velocidad del vehículo y el tiempo continuo para pitar, así como el cupo de tiempo diario acumulable para pitar. Gracias a ello, el producto se puede adaptar a discreción de quien lo use, o en función de la legislación que rija en donde se use. Por defecto, tales variables están definidas para permitir un pitido continuo de hasta una centésima de segundo por cada kilómetro por hora de velocidad; y cinco segundos como máximo tiempo de uso diario.

Para la programación del DLUB se recopilaron e integraron tres librerías obtenidas de la comunidad Arduino, *MsTimer2*, *TimerOne* y *LiquidCrystal*. Como todas las librerías externas de Arduino, estas son escritas por miembros de la comunidad y son para libre disposición de quien las necesite usar. Disponer de programación *open source* facilita de enorme manera la

elaboración de códigos, pues ahorra al programador la escritura de muchas líneas de código que a alguien ya se le ocurrieron previamente.

La plataforma Arduino es una solución sencilla para proyectos sencillos, como aquellos en que basta con incorporar un par de códigos y librerías básicas. A pesar de su gran flexibilidad y opciones para usos avanzados, cuando los diseños se vuelven complejos, es preferible optar por entornos menos simplificados en donde se puede programar con mayor libertad. En el caso del DLUB, se tuvo dificultad para incorporar funciones como la recuperación de tiempo diario para pitar en función del recorrido, o la limitación en el nivel de la bocina en función de la velocidad del vehículo.

Es muy recomendable para ciudades con conductores que abusan del avisador acústico incorporar el DLUB como herramienta para cambiar esta realidad. Esta implementación podría darse como sanción para conductores que fueron amonestados por tal conducta; es decir, si un agente de tránsito multa a un conductor por pitar excesivamente, la sanción para éste debería incluir la adquisición y uso del dispositivo por un tiempo determinado, un año, por ejemplo. Tal sanción tendría gran potencial reformativo, ya que el dispositivo permite al conductor estar consciente de cuánto ha usado el avisador acústico, además de que la autoridad contaría con un instrumento que le proporcione evidencia ante sanciones.

El Dispositivo Limitador del Uso de la Bocina podría incorporarse también a los vehículos de emergencia autorizados para la utilización de sirenas, de modo que los conductores deban justificar mediante bitácora cada vez que hayan encendido su sirena, tomando privilegios de tránsito reservados para emergencias reales. Esto ayudaría a mejorar la credibilidad de instituciones cuyos vehículos de emergencia, es de conocimiento popular, se abren paso entre el tráfico de manera injustificada, y así aumentar la predisposición colectiva de cederles el paso.

El DLUB se perfila como una valiosa herramienta para que entidades gubernamentales y demás interesados en la reducción del ruido de tráfico puedan regular el uso de la bocina. Tal es así que el Departamento de Sanidad de la ciudad de Nueva York expresó su interés en conocer a detalle el prototipo, ya que el dispositivo creado ayuda a evitar efectivamente el abuso de la bocina, tanto mediante la interrupción de su funcionamiento, como a través de la medición y registro diario de su uso. A esto hay que añadirle que, en su versión final, el aparato no debería superar los USD 100 de costo de elaboración, por lo que resultaría una solución de bajo costo con un gran rédito social y medio ambiental.

REFERENCIAS

- Berglund, B., Lindvall, T., & Schwela, D. H. (1999). *Guidelines for community noise. Guidelines for community noise*. OMS.
- Brautigam, S. (1994). *Rethinking the Regulation of Car Horn and Car Alarm Noise: An Incentive Based Proposal to Help Restore Civility to Cities*. *Columbia Journal of Environmental Law*, 19 (391).
- Den Boer, L. C., & Schroten, A. (2007). *Traffic noise reduction in Europe*. *CE Delft*, 14, 2057-2068.
- Evans, B. (2011). *Beginning Arduino Programming*. Denver, Estados Unidos. Apress.
- Evans, M., Noble, J. J., & Hochenbaum, J. (2013). *Arduino in action*. New York: Manning.
- Fernández, P. L. (2000). Conceptos físicos de las ondas sonoras. Física y Sociedad, Revista del Colegio Oficial de Físicos (11).
- Fortino, A., Eckstein, L., Viehöfer, J., and Pampel, J., (2016) *Acoustic Vehicle Alerting Systems (AVAS) - Regulations, Realization and Sound Design Challenges*. *SAE International Journal of Passenger Cars - Mechanical Systems* 9 (3). doi: 10.4271/2016-01-1784
- Goswami, S. (2009). *Road traffic noise: A case study of Balasore town, Orissa, India*. *International Journal of Environmental Research*, 3 (2).
- Grad, F.P., Rosenthal, A.J., Rockett, L.R., Fay, J.A., Heywood, J., Kain, J.F., Ingram, G.K., Harrison, D. Jr., & Tietenberg, T. (1975). *Automobile and the regulation of its impact on the environment*. Estados Unidos.
- Jee, S., Kim, M., Bae, M. (2017a). *On designing a new sound of the car-horn*. *The Journal of the Acoustical Society of America*, 141 (3494).
- Jee, S., Kim, M., Bae, M. (2017b). *A study on a friendly automobile klaxon production with rhythm*. *The Journal of the Acoustical Society of America*, 142 (2507).

- Kang, H. S., Shin, T., Lee, S. K., & Park, D. C. (2015). *Design optimization of a dual-shell car horn for improved sound quality based on numerical and experimental methods. Applied Acoustics*, 90.
- Kudo, M., Fukami, T., & Hayashibe, R. (1988). *Application of a piezoelectric flexural vibrator to a car horn. The Journal of the Acoustical Society of America*, 84 (1).
- Kwon Lee, S., Uk Han, M. (2017). *Observation and evaluation model of warning sound in an electric vehicle warning index based on whine index. The Journal of the Acoustical Society of America*, 141 (3877).
- Lemaitre, G., Susini, P., Winsberg, S., McAdams, S., & Letinturier, B. (2007). *The sound quality of car horns: a psychoacoustical study of timbre. Acta acustica united with Acustica*, 93 (3).
- Ley Orgánica de Transporte Terrestre, Tránsito y Seguridad Vial. Registro Oficial Suplemento 398, Quito, Ecuador, 31 de diciembre de 2014.
- Melgar, E. R., & Diez, C. C. (2012). *Arduino and kinect projects: Design, build, blow their minds*. Montreal, Canadá. Apress.
- Miyara, F. (1999). Control de ruido. Recuperado el 15 de enero de 2018 de:
<http://www.ingenieroambiental.com/4023/control%20de%20ruido,federico%20miyara.pdf>
- Misdariis, N., Gruson, A., Susini, P. (2013). *Detectability study of warning signals in urban background noises: A first step for designing the sound of electric vehicles. Proceedings of Meetings on Acoustics*, 19 (040032), 1. doi: 10.1121/1.4799454
- Noble, J. (2009). *Programming Interactivity: A Designer's Guide to Processing, Arduino, and Openframeworks*. Sebastopol, Estados Unidos. O'Reilly Media, Inc.

Omidvari, M., Nouri, J. (2009). *Effects of noise pollution on traffic policemen. International Journal of Environmental Research*, 3 (4).

Ordenanza Metropolitana 0213 “De la Prevención y Control del Medio Ambiente”. Libro II del Código Municipal, Quito, Ecuador, 18 de Abril de 2007.

Ortega, J. (2014). Mal uso de la bocina, una infracción común. El Comercio. Recuperado de: <http://www.elcomercio.com/actualidad/seguridad/mal-de-bocina-infraccion-comun.html>

Parizet, E., Robart, R., Chamard, J., Schlittenlacher, J., Pondrom, P., Ellermeier, W., Biancardi, F., Janssens, K., Speed-Andrews, P., Cockram, J., Hatton, G. (2013). *Detectability and annoyance of warning sound for electric vehicles. Proceedings of Meetings on Acoustics*, 19 (040033).

Peeters, B., Ammeriaan, I., Kuijpers, A., & van Blokland, G. (2010). *Reduction of the horn effect for car and truck tyres by sound absorbing road surfaces. INTER-NOISE and NOISE-CON Congress and Conference Proceedings*, 2010 (11). Institute of Noise Control Engineering.

Sandoval, A. M. (2005). Ruido por tráfico urbano: conceptos, medidas descriptivas y valoración económica. *Revista de economía y administración*, Universidad Autónoma de Occidente, 2 (1).

Reglamento a la Ley Orgánica de Transporte Terrestre, Tránsito y Seguridad Vial. Registro Oficial Suplemento 731, Quito, Ecuador, 14 de noviembre de 2016.

Reglamento Técnico Ecuatoriano 034 (4R) “Elementos mínimos de seguridad en vehículos automotores”. Resolución No. 16 382 INEN, Quito, Ecuador, 28 de octubre de 2016.

Smith, A. G. (2011). *Introduction to Arduino. A piece of cake*. Recuperado el 10 de diciembre de 2017 de:

<http://www.introtoarduino.com/downloads/IntroArduinoBook.pdf>

- Tu, C. Y., Kuo, W. C., Teng, W. H., Wang, Y. T., & Shiau, S. (2010). *A power-aware cloud architecture with smart metering. In Parallel Processing Workshops (ICPPW), 2010 39th International Conference on.* IEEE.
- Uk Han, M., Kwon Lee, S. (2017). *Warning sound generation system of an electric vehicle system. The Journal of the Acoustical Society of America* 141 (3883).
- Yamauchi, K. (2016). *Effectiveness of additional warning sounds for quiet vehicles in urban noise environment. The Journal of the Acoustical Society of America*, 140 (3433). doi: 10.1121/1.4971059
- Young Gwak, D., Yoon, K., Seong, Y. (2014). *Application of subharmonics for active sound design of electric vehicles. The Journal of the Acoustical Society of America* 136 (EL391).

ANEXOS

ANEXO 1. CÓDIGO FUENTE DE APLICACIÓN ARDUINO DLUB.

1. Inicialización de librerías, pines y variables.

```
#include <MsTimer2.h>
#include <TimerOne.h>
#include <LiquidCrystal.h>

const float secPerKilometerToLimit = 0.01;
const float maxTimeClaxonOnAtDay = 5.0;

const int alarmPinOrange = 5;
const int alarmPinRed = 6;
const int claxonPin = 2;

const float maxSpeed = 200.0;
float maxSpeedADCValue = 0;

int speedSensorPin = A0;
int claxonSensorPin = A1;
int speedSensorValue = 0;
int claxonSensorValue = 0;

LiquidCrystal lcd(12, 11, 10, 9, 8, 7);

int msec = 0;
int sec = 0;
int min_cont = 0;
int hour_cont = 0;
int day = 0;
int month = 0;
int year = 0;

int pos = 8;
int pos2 = 0;

int memory[31][9] = {};
int memoryCont = 0;

boolean flag = 0;

float totalSecClaxonOn = 0.0;
```

```
float totalSecClaxonOn2 = 0.0;
float totalTimeAtDay = 0.0;
float percentClaxonOnAtDay = 0.0;
float speedLimitInSec = 0.0;
int speedValueToLimitClaxon = 0;
boolean flagDataOfDaySavedOnMemory = false;
const int buttonPin = 4;
```

2. Configuración de inicio.

```
void setup(void)
{
    maxSpeedADCValue = (maxSpeed + 1) / 1024.0;

    initDateTime(16, 12, 5, 23, 55, 50);
    //Year Month Day Hour Min Sec

    lcd.begin(16, 2);

    pinMode(alarmPinOrange, OUTPUT);
    pinMode(alarmPinRed, OUTPUT);
    pinMode(claxonPin, OUTPUT);
    pinMode(buttonPin, INPUT);

    Timer1.initialize(100000);
    Timer1.attachInterrupt(ISR_Clock);
    Serial.begin(9600);

    MsTimer2::set(100, getADCValues);
    MsTimer2::start();
}
```

3. Actualización de información en el LCD

```
void ISR_Clock()
{
    update_clock();
}
```

```
setLCDClock();
setClaxonTimeInfoLCD();}
```

4. Lectura de pines analógicos

```
void getADCValues() {
    speedSensorValue = analogRead(speedSensorPin);
    claxonSensorValue = analogRead(claxonSensorPin);
    speedSensorValue = (int)((float)speedSensorValue * maxSpeedADCValue);
    setSpeedSensorValueOnLCD();
}
//Casting Conversión de entero a flotante, para normalizar
```

5. Impresión de la información de tiempo en el LCD.

```
void setClaxonTimeInfoLCD() {

    String info = (String)totalTimeAtDay + "seg " +
    (String)percentClaxonOnAtDay + "%";
    lcd.setCursor(0, 1);
    lcd.print(info);

    if (percentClaxonOnAtDay >= 90.0 and percentClaxonOnAtDay < 100.0) {
        digitalWrite(alarmPinOrange, HIGH);
    }

    if (percentClaxonOnAtDay >= 100.0) {
        digitalWrite(alarmPinOrange, LOW);
        digitalWrite(alarmPinRed, HIGH);

        if (flagDataOfDaySavedOnMemory == false) {

            flagDataOfDaySavedOnMemory = true;
            setDataMatrix(memoryCont);
            memoryCont++;
        }
    }
}
```

6. Registro de infracción en memoria.

```
void setDataMatrix(int i) {
  memory[i][0] = year;
  memory[i][1] = month;
  memory[i][2] = day;
  memory[i][3] = hour_cont;
  memory[i][4] = min_cont;
  memory[i][5] = sec;
  memory[i][6] = totalTimeAtDay;
  memory[i][7] = percentClaxonOnAtDay;
  memory[i][8] = speedValueToLimitClaxon;
}
```

7. Mostrar valor de velocidad en el LCD

```
void setSpeedSensorValueOnLCD() {
  lcd.setCursor(pos2, 0);
  lcd.print(" Km/h");
  lcd.setCursor(pos2, 0);
  lcd.print(speedSensorValue);
}
```

8. Ubicación del reloj en el LCD

```
void setLCDClock() {

  lcd.setCursor(pos, 0);
  lcd.print(" : :");

  lcd.setCursor(pos + 6, 0);
  lcd.print(sec);

  lcd.setCursor(pos + 3, 0);
  lcd.print(min_cont);

  lcd.setCursor(pos, 0);
  lcd.print(hour_cont);
}
// poner datos de reloj en lcd
```

9. Inicialización del calendario

```
void initDateTime(int y, int m, int d, int h, int mi, int se) {  
    year = y;  
    month = m;  
    day = d;  
    hour_cont = h;  
    min_cont = mi;  
    sec = se;  
}
```

10. Reloj

```
void update_msec() {  
    sec++;  
    msec = 0;  
}  
  
void update_sec() {  
    lcd.setCursor(pos + 7, 0);  
    lcd.print(" ");  
    min_cont++;  
    sec = 0;  
}  
  
void update_min() {  
    lcd.setCursor(pos + 4, 0);  
    lcd.print(" ");  
    hour_cont++;  
    min_cont = 0;  
}  
  
void update_hour() {  
    lcd.setCursor(pos + 1, 0);  
    lcd.print(" ");  
    day++;  
    hour_cont = 0;  
    resetValues();  
}
```

```
void update_day() {  
    lcd.setCursor(pos + 7, 1);  
    lcd.print(" ");  
    month++;  
    day = 0;  
}
```

```
void update_month() {  
    lcd.setCursor(pos + 4, 1);  
    lcd.print(" ");  
    year++;  
    month = 0;  
}
```

```
void update_year() {  
    lcd.setCursor(pos + 1, 1);  
    lcd.print(" ");  
    year = 0;  
}
```

11. Función complementaria de reloj

```
void update_clock() {  
    msec++;  
    if (msec == 10) {  
        update_msec();  
    }  
    if (sec == 60) {  
        update_sec();  
    }  
    if (min_cont == 60) {  
        update_min();  
    }  
    if (hour_cont == 24) {  
        update_hour();  
    }  
    if (day == 30) {  
        update_day();  
    }  
    if (month == 12) {  
        update_month();  
    }  
    if (year == 3000) {  
        update_year();  
    }  
}
```

```
}  
}  
}  
}  
}  
}  
}
```

12. Reinicio de valores

```
void resetValues() {  
  
    if (flagDataOfDaySavedOnMemory == false) {  
        flagDataOfDaySavedOnMemory = true;  
        setDataMatrix(memoryCont);  
        memoryCont++;  
    }  
    flag = 0;  
  
    digitalWrite(alarmPinOrange, LOW);  
    digitalWrite(alarmPinRed, LOW);  
  
    totalSecClaxonOn = 0.0;  
    totalSecClaxonOn2 = 0.0;  
  
    totalTimeAtDay = 0.0;  
  
    percentClaxonOnAtDay = 0.0;  
  
    speedLimitInSec = 0.0;  
    speedValueToLimitClaxon = 0;  
  
    flagDataOfDaySavedOnMemory = false;  
  
    lcd.clear();  
}
```

13. Bucle

```
void loop(void)
```

```

{
  if (digitalRead(buttonPin) == HIGH) {
    Serial.println("*****Registro de Infracciones*****");
    for (int i; i < 32; i++) {
      if (memory[i][2] > 0 and memory[i][2] < 40) {
        String infoSerial = "Date: " + (String)memory[i][2] + "/" _
          " + (String)memory[i][1] + " / " + (String)memory[i][0] + " Time: " _
            " + (String)memory[i][3] + ":" + (String)memory[i][4] + ":" _
              " + (String)memory[i][5];
          infoSerial = infoSerial + " Total Time: " + (String)memory[i][6] + "
seconds Percent:
          +(String)(String)memory[i][7] + " %" + " Speed: " +
(String)memory[i][8] + " Km/h";
        Serial.println(infoSerial);
      }
    }

    Serial.println("*****");
    delay(300); // para facilitar lectura del pulsador en digital
  }

  if ((claxonSensorValue > 300) and (flag == 0 ) ) {
    flag = 1;
    Serial.print("Claxon On! ");
    totalSecClaxonOn = (float)min_cont * 60.0 + (float)sec + (float)msec / 10.0 ;
    speedValueToLimitClaxon = speedSensorValue;
    speedLimitInSec = speedValueToLimitClaxon * secPerKilometerToLimit;
    digitalWrite(claxonPin, HIGH);
  }

  if ((claxonSensorValue > 300) and (flag == 1 ) ) {
    totalSecClaxonOn2 = ((float)min_cont * 60.0 + (float)sec + (float)msec /
10.0);
    totalSecClaxonOn2 = totalSecClaxonOn2 - totalSecClaxonOn;
    if (totalSecClaxonOn2 > speedLimitInSec) {
      digitalWrite(claxonPin, LOW);
    }
  }

  if ((claxonSensorValue < 300) and (flag == 1 ) ) {
    digitalWrite(claxonPin, LOW);
  }
}

```

```
flag = 0;
Serial.println("Claxon Off! ");
totalSecClaxonOn = ((float)min_cont * 60.0 + (float)sec + (float)msec /
10.0) - totalSecClaxonOn;

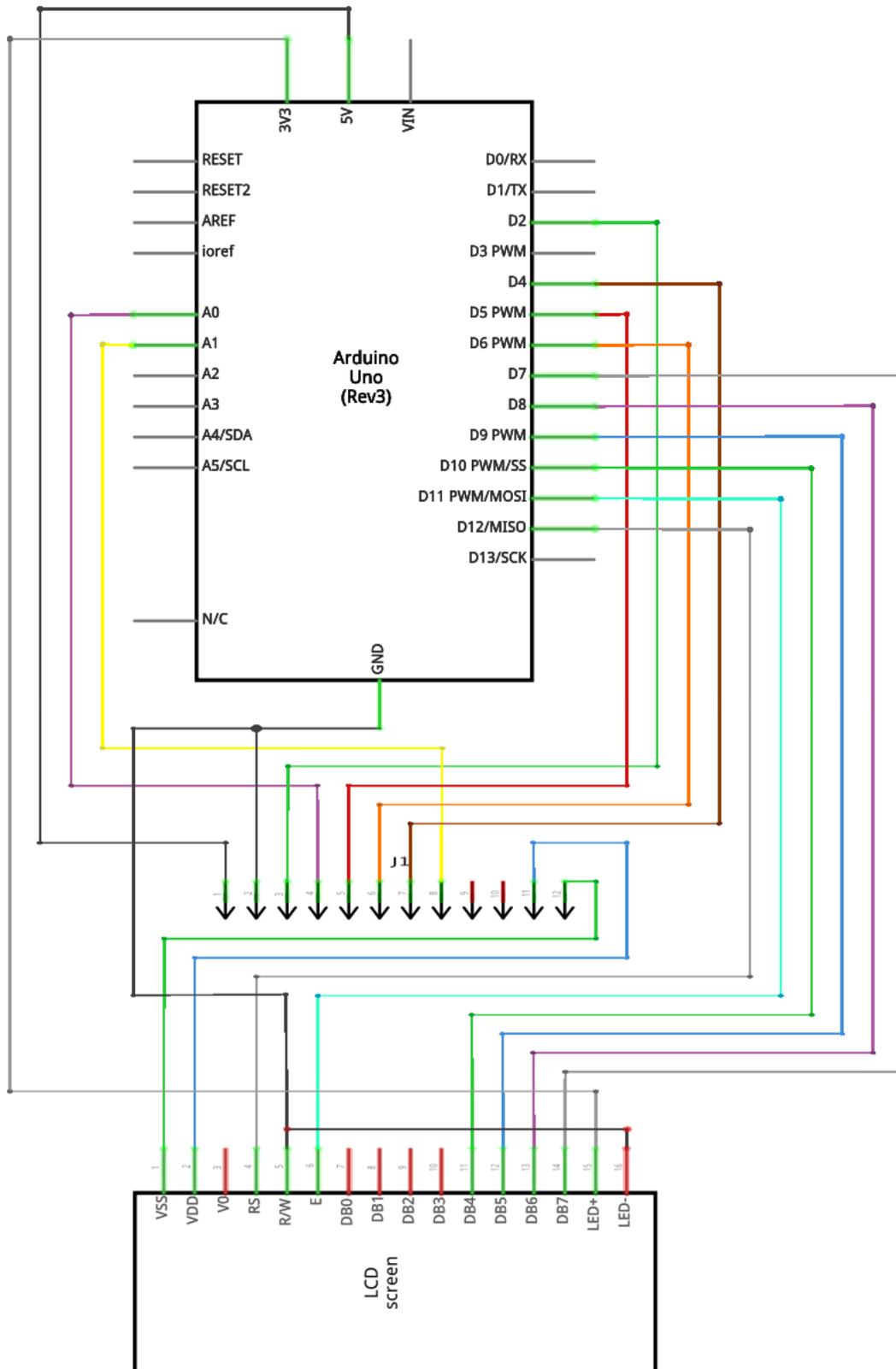
if (totalSecClaxonOn2 > speedLimitInSec) {
    totalTimeAtDay = totalTimeAtDay + speedLimitInSec;
}
else {
    totalTimeAtDay = totalTimeAtDay + totalSecClaxonOn;
}
percentClaxonOnAtDay = (totalTimeAtDay / maxTimeClaxonOnAtDay) *
100;
}
delay(1);
}
```

ANEXO 2. IMÁGENES GENERALES.

1. DLUB conectado a ordenador con IDE Arduino.



2. Diagrama de conexión entre placa Arduino y LCD.



ANEXO 3. MATERIAL AUDIOVISUAL.

[Breve explicación del funcionamiento del DLUB.](https://youtu.be/T2IN34rIMTE)

<https://youtu.be/T2IN34rIMTE>



[Video Blog de concienciación sobre el uso de la bocina, presentando el DLUB.](https://youtu.be/b3M6Lnijz1s)

<https://youtu.be/b3M6Lnijz1s>



