

Universidad Internacional del Ecuador



Facultad de ingeniería mecánica automotriz

**Proyecto de grado para la obtención del Título de Ingeniería en Mecánica
Automotriz**

**Diseño y construcción de un módulo medidor de velocidad y aceleración
del vehículo por Arduino y GPS**

De la A León Kevin Israel

Director: Ing. Edwin Puente

Guayaquil, abril 2018

UNIVERSIDAD INTERNACIONAL DEL ECUADOR

FACULTAD DE INGENIERÍA MECÁNICA AUTOMOTRIZ

CERTIFICADO

Ing. Edwin Puente

CERTIFICA:

Que el trabajo titulado **“DISEÑO Y CONSTRUCCIÓN DE UN MÓDULO MEDIDOR DE VELOCIDAD Y ACELERACIÓN DEL VEHÍCULO POR ARDUINO Y GPS”**, realizado por el estudiante: **KEVIN ISRAEL DE LA A LEÓN**, ha sido guiado y revisado periódicamente y cumple las normas estatutarias establecidas por la Universidad Internacional del Ecuador, en el Reglamento de Estudiantes.

Debido a que constituye un trabajo de excelente contenido científico que coadyuvará a la aplicación de conocimientos y al desarrollo profesional. Si recomiendo su publicación. El mencionado trabajo consta de (un) empastado y (un) disco compacto el cual contiene los archivos en formato portátil de Acrobat. Autoriza al señor: Kevin Israel De La A León, que lo entregue a biblioteca de la Facultad, en su calidad de custodia de recursos y materiales bibliográficos.

Guayaquil, abril 2018



Ing. Edwin Puente

Director del proyecto

UNIVERSIDAD INTERNACIONAL DEL ECUADOR

FACULTAD DE INGENIERÍA EN MECÁNICA AUTOMOTRIZ

Yo, KEVIN ISRAEL DE LA A LEÓN

DECLARO QUE:

La investigación de cátedra denominada: **“DISEÑO Y CONSTRUCCIÓN DE UN MÓDULO MEDIDOR DE VELOCIDAD Y ACELERACIÓN DEL VEHÍCULO POR ARDUINO Y GPS”**, ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de mi autoría, apoyados en la guía constante de mi docente.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico para la Facultad de Ingeniería en Mecánica Automotriz.

Guayaquil, abril 2018.



Kevin Israel De La A León

C.I: 0930296579

ÍNDICE GENERAL

CERTIFICADO	iii
DECLARACIÓN DE RESPONSABILIDAD	iv
AUTORIZACIÓN	v
ÍNDICE GENERAL	vi
DEDICATORIA	xiii
AGRADECIMIENTO	xiv
RESUMEN GENERAL.....	xv
ABSTRACT	xvi
INTRODUCCIÓN.....	xvii
CAPÍTULO I.....	1
PROBLEMA DE LA INVESTIGACIÓN Y MARCO REFERENCIAL.....	1
1.1.Definición del problema	1
1.2.Objetivos de la investigación	2
1.2.1.Objetivo general.....	2
1.2.2.Objetivos específicos	2
1.3.Alcance.....	2
1.4.Justificación e importancia de la investigación	3
1.5.Marco Metodológico	3
1.5.1.Método de investigación	3
1.5.2.Tipo de investigación	4
1.6.Marco teórico.....	4
1.6.1.¿Qué es la electricidad?	4
1.6.2.¿Qué es el voltaje?	6
1.6.3.¿Qué es la intensidad de corriente?	7
1.6.4.Ley de Ohm	8
1.6.5.La potencia	9
1.6.6.Circuitos eléctricos.....	10
1.6.7.Conexiones en serie y en paralelo	11
1.6.8.Sistema electrónico	13

1.6.9.Microcontrolador MAX7219.....	14
1.6.10.CPU	15
1.6.11.Memorias	16
1.6.12.Entradas y salidas	16
1.6.13.Lenguaje de programación	18
1.6.14.Arduino	19
1.6.15.GPS	21
CAPÍTULO II.....	23
ELEMENTOS DEL MODULO ARDUINO	23
2.1.Componentes y datos técnicos	23
2.2.Pines de alimentación	25
2.3.Entradas y salidas digitales	26
2.4.Entradas analógicas	27
2.5.Las salidas analógicas.....	28
2.6.Software Arduino	29
2.7.El reloj o metrónomo.....	31
2.8.El botón de reseteo.....	32
CAPÍTULO III.....	33
HARDWARE Y SOFTWARE PARA EL DISEÑO Y CONSTRUCCIÓN DE ARDUINO Y GPS	33
3.1.Diseño del módulo	33
3.2.Implementación de hardware y software de Arduino.....	33
3.3.Construcción del Arduino	34
3.4.Instalación del GPS	37
3.5.Visualización de datos GPS en Arduino.....	41
3.6.Programación de hora	42
3.7.Velocímetro	44
3.8.Tarjeta de memoria.....	44
CAPÍTULO IV	48
COMPROBACIÓN Y RESULTADOS DE UN MÓDULO MEDIDOR DE VELOCIDAD Y ACELERACIÓN DEL VEHÍCULO POR ARDUINO Y GPS.....	48

4.1.Comprobación y funcionamiento del módulo	48
4.2.Modo de emplearse	48
4.3.Instalación de antena GPS en el vehículo.....	49
4.4.Revision de datos en la memoria SD	50
4.5.Registro de datos en el programa.....	51
4.6. Programación	52
4.7.Curva gráfica de prueba	54
CAPÍTULO V	55
CONCLUSIONES Y RECOMENDACIONES	55
5.1.Conclusiones	55
5.2.Recomendaciones	56
BIBLIOGRAFÍA.....	57
ANEXOS	58
GLOSARIO.....	59

ÍNDICE DE FIGURAS

Figura 1. Electrones.....	5
Figura 2. Ámbar.....	5
Figura 3. Voltaje.....	6
Figura 4. Intensidad.....	7
Figura 5. Ley de Ohm.....	8
Figura 6. Potencia consumida.....	9
Figura 7. Microprocesador MAX7219.....	14
Figura 8. Descripción del microprocesador.....	15
Figura 9. CPU.....	15
Figura 10. Memoria.....	16
Figura 11. Lenguaje de Programación C++.....	18
Figura 12. Software de aplicaciones.....	19
Figura 13. Arduino conexión.....	20
Figura 14. GPS.....	21
Figura 15. Navegación GPS.....	22
Figura 16. Microcontrolador ATmega328p.....	23
Figura 17. Arduino UNO.....	24
Figura 18. Pines de alimentación.....	26
Figura 19. Pines de entrada y salida.....	26
Figura 20. Entradas analógicas.....	27
Figura 21. Software Arduino.....	29
Figura 22. Programación.....	30
Figura 23. Metrónomo.....	31
Figura 24. Botón reset.....	32
Figura 25. Conexión pantalla LCD.....	34
Figura 26. Placa GPS.....	34
Figura 27. Conexión al módulo GPS.....	35
Figura 28. Cableado al módulo GPS.....	36
Figura 29. Interruptor GPS.....	36

Figura 30. Señal GPS	37
Figura 31. Archivo de datos	39
Figura 32. Ingreso de datos GPS.....	40
Figura 33. Ingreso google maps	40
Figura 34. Ingreso datos Arduino.....	41
Figura 35. Conexiones de Max 7219	44
Figura 36. Conexión de tarjeta SD	45
Figura 37. Tarjeta SD	45
Figura 38. Registro velocidad	46
Figura 39. Monitoreo GPS	47
Figura 40. Montaje en el vehículo.	48
Figura 41. Instalación de antena GPS.	49
Figura 42. Montaje antena GPS.....	49
Figura 43. Alimentación 9v.	50
Figura 44. Archivo .txt.....	51
Figura 45. Registro de datos.....	51
Figura 46. Ubicación GPS	52
Figura 47. Programación	53
Figura 48. Curva gráfica de la velocidad del vehículo	54

ÍNDICE DE TABLAS

Tabla 1. Elementos del Módulo Arduino (Exploring Arduino)	25
Tabla 2. Pines de alimentación (Exploring Arduino).....	25
Tabla 3. Entradas y salidas digitales (Exploring Arduino)	27
Tabla 4. Códigos del GPS (Exploring Arduino)	38
Tabla 5. Horario (Geek Factory)	42

ÍNDICE DE DIAGRAMAS

Diagrama 1. Circuito eléctrico	11
Diagrama 2. Conexión serie y paralelo	12
Diagrama 3. Sistema Eléctrico	13
Diagrama 4. Entradas y salidas	17
Diagrama 5. Salidas analógicas.....	28
Diagrama 6. Horario	43

DEDICATORIA

Dedico este proyecto de investigación a mi Familia, a mi Dios, quienes me han guiado por el buen camino, dándome todas las fuerzas para seguir adelante enfrentando los obstáculos sin perder nunca la dignidad.

A mis padres por su apoyo, consejos, comprensión, amor, ayuda en los momentos difíciles, y por ayudarme con los recursos necesarios para estudiar. Me han ayudado a mejorar como persona, inculcándome valores y principios, que han servido para moldear mi carácter, empeño, perseverancia, y coraje para conseguir mis objetivos.

A mis maestros y amigos; a mi tutor de proyecto y a todos quienes han recorrido conmigo este camino, porque me han enseñado a ser más humano.

KEVIN ISRAEL DE LA A LEÓN

AGRADECIMIENTO

La concepción de este proyecto está dedicada a mis padres, pilares fundamentales en mi vida. Sin ellos, jamás hubiese podido conseguir lo que hasta ahora tengo. Su tenacidad y lucha insaciable han hecho de ellos un gran ejemplo a seguir y destacar, no solo para mí, sino para mis hermanos y familia en general.

KEVIN ISRAEL DE LA A LEÓN

RESUMEN GENERAL

La mayoría de los conductores conducen más rápido de lo que recomienda el límite de velocidad, por lo general superan tal límite en más de 100 km/h. El exceso de velocidad es un riesgo de sufrir un accidente y aumentan las probabilidades de sufrir lesiones graves o de muerte. Pero también por conductores que rebasan estos límites de velocidad perjudicando a quienes al momento de pasar el foto radar están uno al lado de otro en cada carril, causando una multa para los 2 vehículos.

El cumplimiento de la ley sigue siendo un elemento importante para los infractores. Las medidas tradicionales para asegurar el cumplimiento incluyen vigilantes de tránsito en las carreteras y foto radares. Gracias a nuevas tecnologías hay formas de identificar cada vehículo, el control inteligente de límites de velocidad y en el futuro será posible asegurar el cumplimiento en todas partes y en todo momento.

El presente trabajo, está basado en el diseño y construcción de un módulo medidor de velocidad y aceleración que registra cada segundo la ubicación exacta y la velocidad que anduvo el vehículo, ayudando a los conductores a gestionar una apelación ante una multa injusta que haya tenido al momento de cruzar el foto radar.

ABSTRACT

Most drivers drive faster than the speed limit recommends, they usually exceed that limit by more than 100 km / h. Speeding is a risk of an accident and increases the chances of serious injury or death. But also by drivers who exceed these speed limits harming those who when passing the photo radar are next to each other in each lane, causing a fine for the 2 vehicles.

Compliance with the law remains an important element for offenders. Traditional measures to ensure compliance include road traffic guards and photo radars. Thanks to new technologies there are ways to identify each vehicle, the intelligent control of speed limits and in the future it will be possible to ensure compliance everywhere and at all times.

The present work is based on the design and construction of speed and acceleration measuring module that registers every second the exact location and speed of the vehicle, helping the drivers to file an appeal against an unfair fine at the time to cross the photo radar.

INTRODUCCIÓN

La invención y la creación del automóvil es una prioridad en nuestras vidas hoy. La industria produce automóviles para ayudar en el movimiento, seguridad y traslado diario siendo una propiedad de gran necesidad e importancia.

El diseño y construcción de un módulo medidor de velocidad y aceleración del vehículo por Arduino y GPS (Global positioning system - Sistema de posicionamiento global) es un accesorio útil, compacto y conveniente en nuestro día a día para poder protegernos en las vías al momento de tener alguna infracción de tránsito por exceso de velocidad que fuese ocasionada por otro conductor.

Este dispositivo se diseñó en base a un sistema de monitoreo mediante registros de velocidades y posicionamiento global para determinar los rangos de velocidad, tiempo y lugar específico que el vehículo realizó por medio de gráficos de curva, para disminuir las multas por exceso de velocidad.

CAPÍTULO I

PROBLEMA DE LA INVESTIGACIÓN Y MARCO REFERENCIAL

1.1. Definición del problema

El problema se centra en que, a diario en la ciudad de Guayaquil vivimos con un gran problema por parte de la Autoridad de Tránsito Municipal (ATM), ya que ellos disponen multas injustificadas, exageradas, innecesarias o erróneas, en lo que respecta a exceso de límites de velocidad.

Los reclamos no se dejan esperar, pues el irrespeto a los límites de velocidad es la principal infracción de tránsito que se comete en la ciudad, según cifras difundidas por la propia entidad municipal. Exceder la velocidad genera multa con \$ 112,50 y 6 puntos menos en la licencia.

Este proyecto de titulación ofrece las herramientas y procesos necesarios que podrán aplicar para el beneficio de los profesionales y estudiantes del sector automotriz, apegándose al objetivo 1 del Plan Nacional de Desarrollo 2017-2021 Toda una Vida, que hace mención a: Garantizar una vida digna con iguales oportunidades para todas las personas, generando conocimiento, motivando y fortaleciendo el desarrollo de sus capacidades técnicas,

obedeciendo las líneas de investigación de la Universidad Internacional del Ecuador: Innovación Tecnológica, Modelación y Simulación de Procesos.

1.2. Objetivos de la investigación

1.2.1. Objetivo general

Diseñar y construir un módulo medidor de velocidad y aceleración del vehículo por Arduino y GPS, para disminuir las multas por exceso de velocidad, que, en muchos de los casos, llegan a ser injustas e impuestas de manera equívoca.

1.2.2. Objetivos específicos

- ✓ Realizar pruebas acerca del módulo medidor de velocidad y aceleración del vehículo.
- ✓ Analizar resultados teóricos y prácticos sobre módulo.
- ✓ Investigar información sobre las funciones del módulo arduino.

1.3. Alcance

El alcance para el análisis de los resultados obtenidos del arduino y GPS son calculados en metros, kilómetros, tiempo, longitud y latitud mediante codificación.

1.4. Justificación e importancia de la investigación

La base teórica del trabajo se fundamenta en temas relacionados al módulo medidor de velocidad y aceleración de vehículos, debido que muchos de los conductores desconocen términos ligados a la mecánica automotriz y es con ellos que se debe de trabajar para profundizar la investigación.

Se usará un sistema mixto de investigación ya que su parte cualitativa se va a ver de forma visual y la cuantitativa se harán cálculos para determinar el tiempo y velocidad del recorrido del vehículo asegurándonos la satisfacción total.

Con Arduino se pueden realizar todo tipo de proyectos de rango muy variado, pasando por sistemas de navegación, sensores ambientales, telemática, etc. Realmente, las posibilidades del arduino para el desarrollo de productos electrónicos son prácticamente infinitas y solo están limitadas por nuestra imaginación.

1.5. Marco Metodológico

1.5.1. Método de investigación

Se considera la aplicación del enfoque de investigación mixto, considerando que de esta manera se podrá expresar de manera porcentual la factibilidad de diseñar y construir un módulo medidor de velocidad y aceleración

del vehículo por Arduino y GPS, en base a la necesidad de respaldo para liberar multas injustas en los casos que lo amerite, por exceso de velocidad.

1.5.2. Tipo de investigación

El módulo medidor de velocidad y aceleración de vehículos por Arduino y GPS, ayudará a que cada conductor pueda grabar y controlar la velocidad al momento de conducir, a la vez que servirá como prueba y soporte en el momento de algún posible reclamo; de la misma manera podrá evaluar la construcción y el funcionamiento del diseño, ya que es necesario conocer sobre las posibles fallas que se podrían presentar, para proceder a tomar la medida correctiva oportuna.

1.6. Marco teórico

1.6.1. ¿Qué es la electricidad?

La electricidad es la tarea que producen los electrones al trasladarse a lo largo de un conductor. La electricidad es una propiedad física de la materia. Consiste en aquella interacción negativa o positiva existente entre los protones y los electrones de la materia, como se muestra en la figura 1.

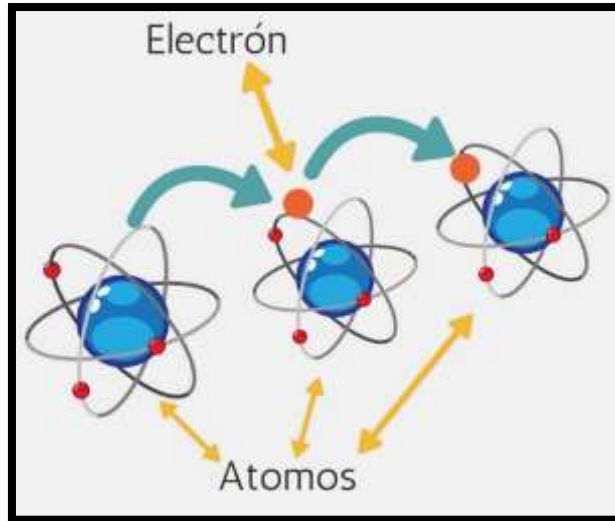


Figura 1. Electrones.
(Energía-nuclear)

El origen etimológico de la palabra es Griego, quienes la estudiaron en esta civilización la llamaron "Ámbar" como se muestra en la figura 2, por el color tan versátil y luminoso que presentaba; sin embargo, el término fue introducido en la sociedad científica por primera vez en Inglaterra para describir el fenómeno de interacción de energía entre partículas.



Figura 2. Ámbar
(Liferder)

1.6.2. ¿Qué es el voltaje?

Es la magnitud física que tiene un circuito eléctrico y que impulsa a los electrones a lo largo de un conductor. Es decir, la energía eléctrica se conduce con mayor o menor potencia debido a que el voltaje es un mecanismo eléctrico entre los dos cuerpos como se muestra en la figura 3.

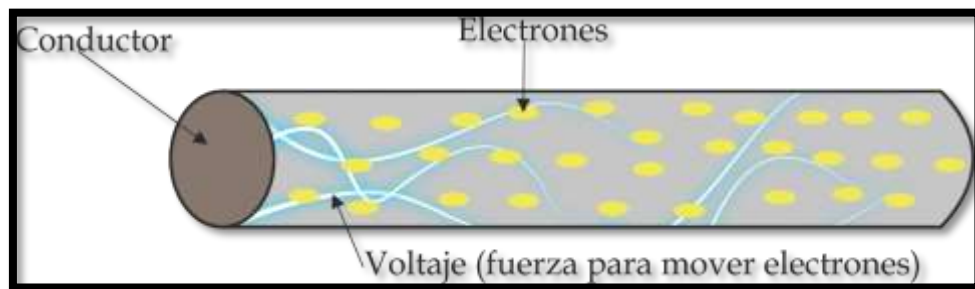


Figura 3. Voltaje
(Espacio Honduras)

El voltaje es una de tensión, es el trabajo por unidad de carga ejercido por el campo eléctrico sobre una partícula para que ésta se mueva de un lugar a otro.

La unidad de medida del es el voltio (V), pero también podemos tener otras unidades de medida como milivoltios ($1 \text{ mV} = 0,001 \text{ V}$), o de kilovoltios ($1 \text{ kV} = 1000 \text{ V}$).

Existen diversos tipos de voltajes que pueden ser, una neurona que tiene un voltaje de (75 mega voltios); la batería o la pila no recargable es de (1,5 voltios); el sistema eléctrico de un automóvil tiene un voltaje de (12 voltios);

la electricidad en las casas es de (230 voltios); en un tren es de (600 a 700 voltios); la red de transporte de electricidad de alto voltaje es de 110 (kilovoltios) y el de un relámpago (100 mega voltios).

1.6.3. ¿Qué es la intensidad de corriente?

Es una magnitud eléctrica que se define como la cantidad de carga eléctrica que pasa a través de un punto concreto de un material conductor en un determinado tiempo. Se puede pensar que la intensidad de corriente es similar en cierto punto al caudal de agua que circula por una tubería la cual pasa más o menos una cantidad de agua por la tubería en un determinado tiempo como se muestra en la figura 4.

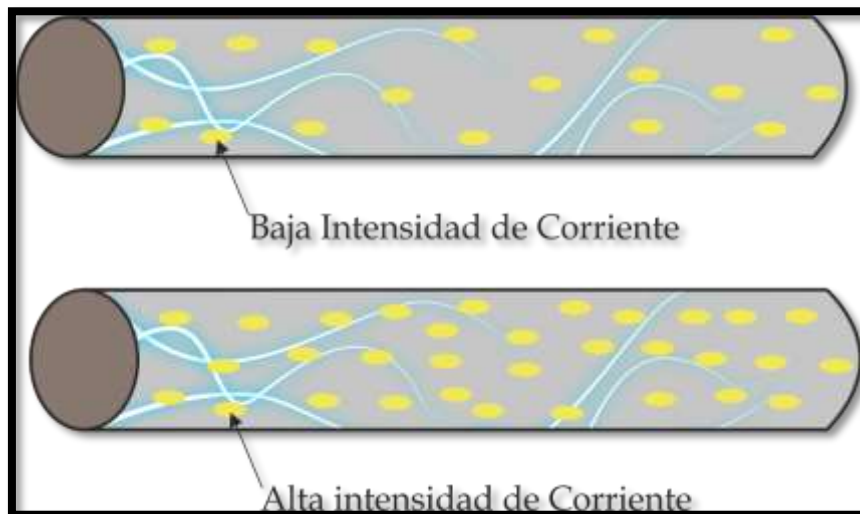


Figura 4. Intensidad
(Espacio Honduras)

Se mide en amperio (A), como también puede ser en miliamperios (1 mA = 0,001 A), de microamperios (1 μ A = 0,001 mA), o incluso de nanoamperios (1 nA = 0,001 μ A).

1.6.4. Ley de Ohm

Nos dice que si un componente eléctrico con resistencia interna, R, es atravesado por una intensidad de corriente en un conductor, I, entre ambos extremos de dicho componente tendremos una diferencia de potencial, V, que puede ser conocida como $V = I \cdot R$, como se muestra en la figura 5.

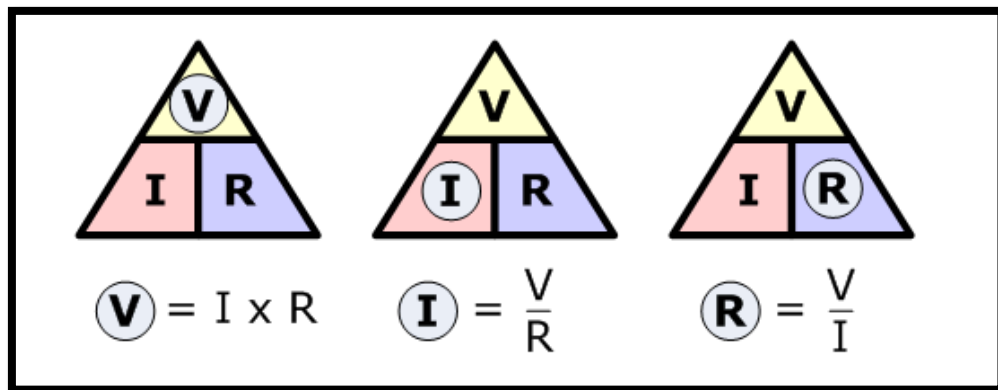


Figura 5. Ley de Ohm
(Fabaacademy)

Con esta fórmula de las tres magnitudes eléctricas es fácil deducir relaciones de proporcionalidad. Por ejemplo: se puede ver que cuanto mayor es la intensidad de corriente, mayor es la diferencia de potencial entre sus extremos. También podemos ver que, cuanto mayor es su resistencia interna, mayor es la diferencia de potencial entre sus dos extremos.

1.6.5. La potencia

Se puede definir la potencia de un componente eléctrico/electrónico, como la energía consumida en un segundo como se muestra en la figura 6, sino, se estaría refiriendo de una fuente de alimentación. Con la palabra potencia se reconoce a la energía eléctrica aportada por ésta al circuito en un segundo. La potencia es un valor intrínseco propio del generador, respectivamente. La unidad de medida de la potencia es el watio (W), pero también podemos verlo en miliwatios ($1 \text{ mW} = 0,001 \text{ W}$), o kilowatios ($1 \text{ kW} = 1000 \text{ W}$).

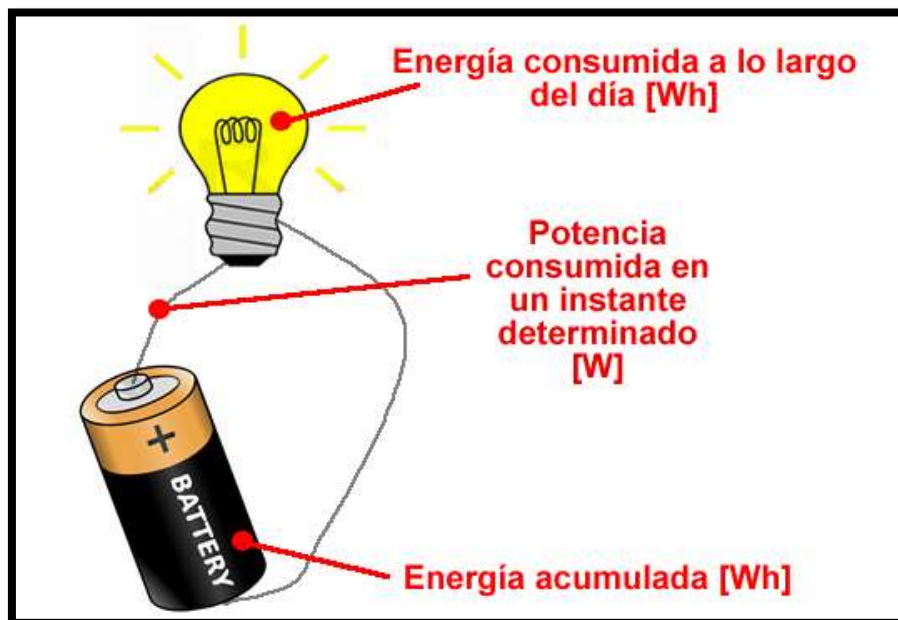


Figura 6. Potencia consumida
(EliseSebast)

A partir de la potencia conocida propia del generador y del tiempo que este esté funcionando, se puede conocer la energía consumida, mediante la expresión: $E = P \cdot t$. (E= energía consumida, P= Potencia, t= tiempo)

Cuando la fuente de alimentación brinda una determinada energía eléctrica, esta puede ser consumida por los distintos componentes del circuito de diversas maneras: la mayoría de veces es gastada en forma de calor debido al efecto de las resistencias internas intrínsecas de cada componente (el llamado “efecto Joule”), pero también puede ser consumida en forma de luz o en forma de movimiento, o en forma de sonido, o en una mezcla de varias.

Se puede calcular la potencia consumida por un componente eléctrico si se tiene el voltaje al que está sometido y la intensidad de corriente que lo atraviesa, utilizando la fórmula $P = V \cdot I$.

1.6.6. Circuitos eléctricos

Para la estructura y la composición de un circuito eléctrico se utilizan esquemas gráficos. Cada dispositivo del circuito se representa mediante un símbolo estandarizado y se dibujan todas las interconexiones existentes entre ellos, como se muestra en el diagrama 1.

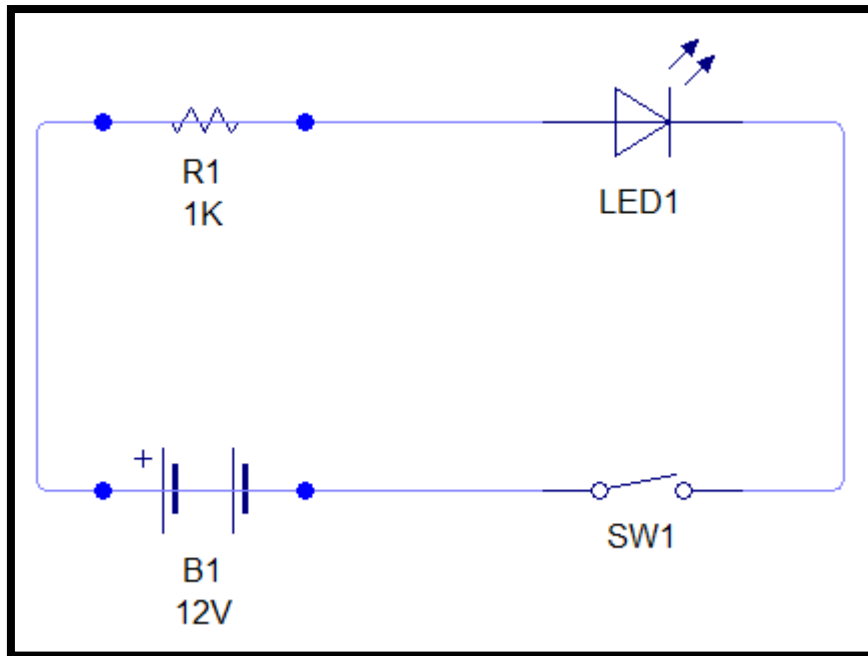


Diagrama 1. Circuito eléctrico
(Kevin de la A.)

En el diagrama anterior podemos apreciar cuatro dispositivos representados por su símbolo convencional: batería, resistencia, un LED (Light emitting Diode - Diodo emisor de luz) y un interruptor.

1.6.7. Conexiones en serie y en paralelo

Los distintos dispositivos presentes en un circuito pueden conectarse entre sí de varias formas. Ya sea en “conexión en serie” y la “conexión en paralelo”.

Si diversos componentes se conectan entre sí en paralelo, a todos ellos se les aplica la misma tensión por igual, trabajando todos al mismo voltaje. Así

mismo, la intensidad de la corriente total será la suma de las intensidades que pasan por cada uno de los componentes, ya que existen varios caminos posibles para el paso de los electrones en el conductor.

Si la conexión es en serie, la tensión total disponible se repartirá entre los diferentes componentes, de manera que cada uno trabaje sometido a una parte de la tensión total. Es decir: la tensión total será la suma de las tensiones en cada componente. Por otro lado, la intensidad de corriente que circulará por todos los componentes en serie será siempre la misma, ya que solo existe un camino posible para el paso de los electrones.

En el diagrama 2, Se puede entender mejor la diferencia en los que se puede ver la conexión en serie y en paralelo de dos resistencias.

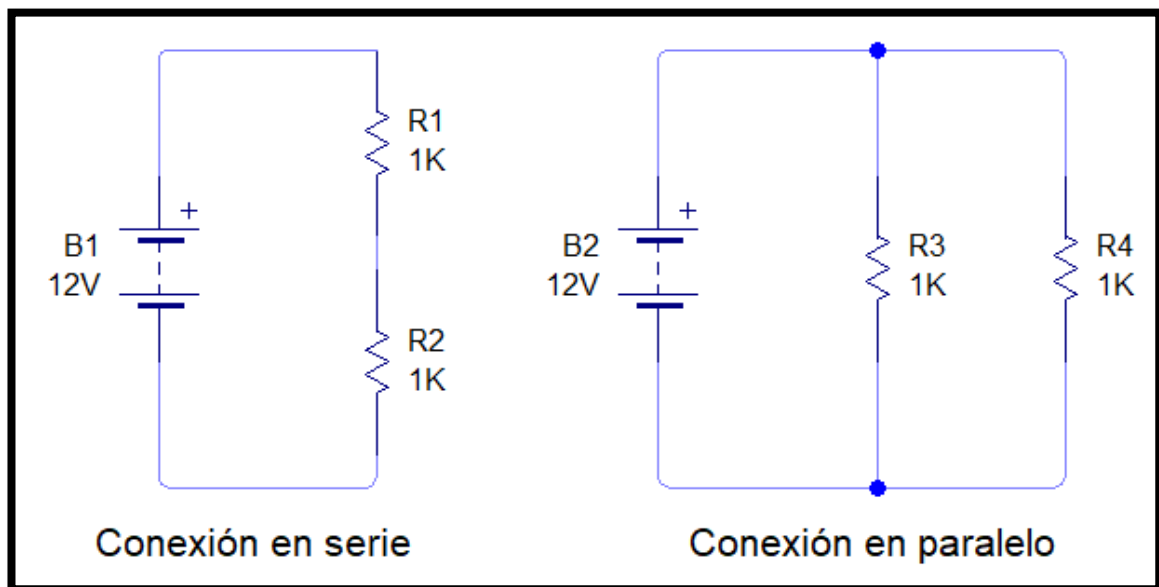


Diagrama 2. Conexión serie y paralelo
(Kevin de la A.)

1.6.8. Sistema electrónico

Un sistema electrónico como se muestra en el diagrama 3, es un conjunto de: sensores, circuitería de procesamiento y control, actuadores y fuente de alimentación.

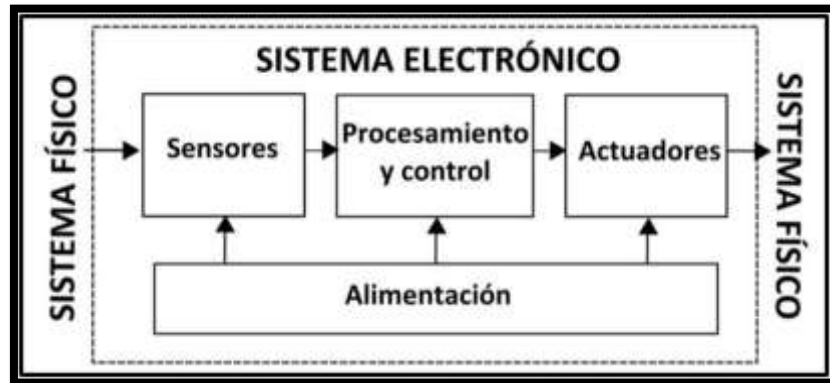


Diagrama 3. Sistema Eléctrico
(Eduar)

Los sensores obtienen información del exterior y la transforman en una señal eléctrica que puede ser manipulada por la circuitería interna de control. Existen sensores de todo tipo: de temperatura, velocidad, magnéticos, infrarrojos, ópticos, de contacto, movimiento de humedad, de sonido (micrófonos), etc.

Los microcontroladores son eléctricos que procesan la señal eléctrica convenientemente. La manipulación de dicha señal dependerá del diseño de los diferentes componentes hardware del sistema, como en el programa que dicho hardware tenga pregrabado y que sea capaz de ejecutar de forma autónoma.

Los actuadores transforman la señal eléctrica acabada de procesar por la circuitería interna en energía que actúa directamente sobre el mundo físico externo. Ejemplos de actuadores son: un motor (energía mecánica), una bombilla (energía lumínica), un altavoz (energía acústica).

1.6.9. Microcontrolador MAX7219

En la figura 7. Un microcontrolador es un circuito integrado que tiene la característica de ser programable. Es decir, que es capaz de ejecutar de forma autónoma una serie de instrucciones previamente definidas por nosotros. En el diagrama 3, representativo de un sistema electrónico, el microcontrolador sería el componente principal de la circuitería de procesamiento y control. En la figura 8 se muestra la descripción del microcontrolador Max7219.

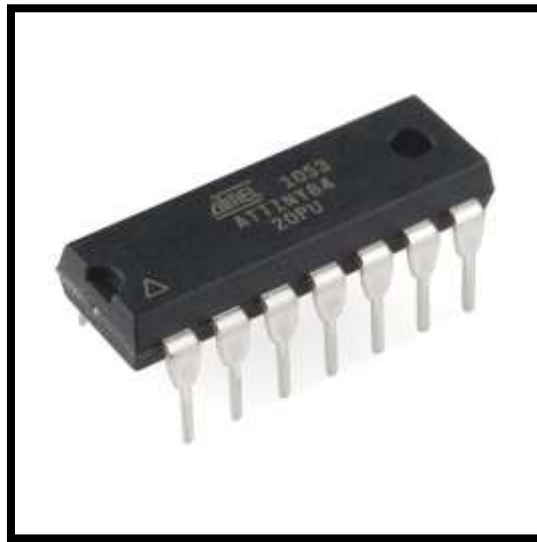


Figura 7. Microprocesador MAX7219
(Teslabem)

entrada bus SPI (de MOSI)	DIN	01	24	DOUT	salida bus SPI (a otro DIN)
dígito 0 del display LED	DIG0	02	23	SEGD	segmento D del display LED
dígito 4 del display LED	DIG4	03	22	SEGP	punto del display LED
0 V	GND	04	21	SEGE	segmento E del display LED
dígito 8 del display LED	DIG8	05	20	SEGC	segmento C del display LED
dígito 2 del display LED	DIG2	06	19	V+	alimentación de 4V a 5V5
dígito 3 del display LED	DIG3	07	18	ISET	referencia de intensidad
dígito 7 del display LED	DIG7	08	17	SEGG	segmento G del display LED
0 V	GND	09	16	SEGB	segmento B del display LED
dígito 5 del display LED	DIG5	10	15	SEGF	segmento F del display LED
dígito 1 del display LED	DIG1	11	14	SEGA	segmento A del display LED
selección de dispositivo SPI	CS	12	13	CLK	entrada de reloj SPI

Figura 8. Descripción del microprocesador
(Polaridad.es)

1.6.10. CPU

En la figura 9. El CPU (Central Processing Unit - Unidad central de proceso) es la parte principal encargada de ejecutar cada instrucción y de controlar que dicha ejecución se cumpla correctamente. Normalmente, estas instrucciones hacen uso de datos disponibles previamente (los “datos de entrada”), y generan como resultado otros datos diferentes (los “datos de salida”), que podrán ser utilizados (o no) por la siguiente instrucción.



Figura 9. CPU
(Shutter stock)

1.6.11. Memorias

Son en general las encargadas de alojar todas las instrucciones como también los diferentes datos que estas necesitan. De esta manera se asegura de que toda esta información esté siempre disponible rápidamente para que la CPU pueda acceder y trabajar con ella en cualquier momento. Como se muestra en la figura 10.

En el mercado se encuentran dos tipos de memorias: las que se almacenan de forma permanente los datos y las que su información se pierde al momento de no tener energía. Según las características de la información a guardar, se grabarán en un tipo de memoria de forma automática.

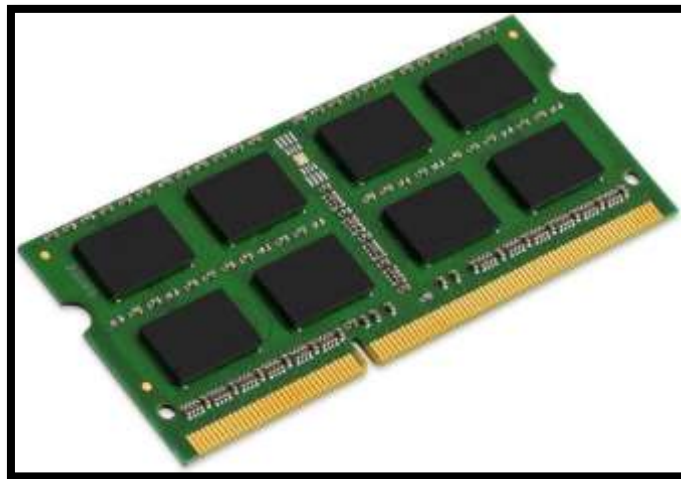


Figura 10. Memoria
(Shutter stock)

1.6.12. Entradas y salidas

Como se muestra en el diagrama 4, son las encargadas de comunicar el microcontrolador con el exterior. En las secciones de entrada del

microcontrolador se conectan sensores para que podamos recibir datos provenientes de su entorno, y en sus secciones de salida se pueden conectar actuadores para enviarles ordenes e interactuar con el medio físico del microcontrolador. De todas formas, muchas secciones de la mayoría de microcontroladores no son solo de entrada o de salida, sino que pueden ser utilizados indistintamente para ambos propósitos.

Un microcontrolador es una computadora completa en un solo chip, el cual está especializado en ejecutar constantemente un conjunto de tareas predefinidas. Estas tareas irán teniendo en cuenta en cada paso que la información es obtenida y enviada por las secciones de E/S (entrada y salida) y reaccionarán en consecuencia. Lógicamente, las tareas serán diferentes según el uso que se le quiera dar al microcontrolador.

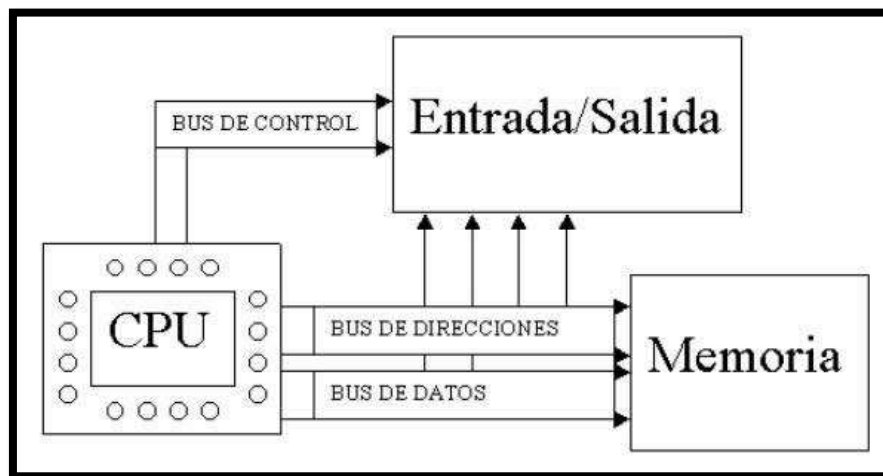


Diagrama 4. Entradas y salidas
(Mindomo)

1.6.13. Lenguaje de programación

Un lenguaje de programación es un lenguaje muy estricto diseñado para describir el conjunto de acciones o instrucciones consecutivas que un computador debe ejecutar. El C ++ es uno de los lenguajes de programación más utilizados hoy en día, se podría decir que es un lenguaje híbrido desarrollado por Bjarne Stroustrup. Como se muestra en la figura 11.

```
int main(int argc, char ** ar
{
    puts("loop test");
    for (int i = 0; argv[i];
        printf("%d: %s\n", i.
    }
    return 0;
```

Figura 11. Lenguaje de Programación C++
(Lynda)

El C ++ es considerado un lenguaje de nivel intermedio, ya que contiene características de lenguaje de alto y bajo nivel. Inicialmente, el lenguaje se llamaba "C con clases" ya que tenía todas las propiedades del lenguaje C con un concepto adicional de "clases". Sin embargo, fue renombrado C ++ en 1983.

El lenguaje de programación C ++ se utilizan principalmente con software de aplicaciones o sistema, controladores, aplicaciones cliente-servidor y firmware incorporado. Como se muestra en la figura 12.



Figura 12. Software de aplicaciones
(Lynda)

El lenguaje de programación incluye varios operadores como comparación, operadores lógicos, manipulación de bits, aritmética y operadores lógicos. Una de las características más atractivas de C ++ es que permite la sobrecarga de ciertos operadores.

1.6.14. Arduino

Arduino nació en el Instituto de interacción y Diseño Ivrea en Italia, es una herramienta fácil de usar, dirigido a estudiantes sin experiencia alguna en electrónica y programación. Tan pronto como llegó a una comunidad más amplia, Arduino comenzó a adaptarse a las nuevas necesidades y desafíos. Todas las placas de Arduino son completamente de código abierto, lo que permite a los usuarios construirlos de forma independiente y eventualmente adaptarlos a sus necesidades particulares. El software que lleva también es de

código abierto y está creciendo a través de las diferentes contribuciones de los usuarios en todo el mundo.

Una de las ventajas son que el software de programación ya viene incorporado, listo para ser utilizado y cuenta con una amplia librería para la programación. Arduino es una plataforma de computación física de lenguaje de programación, basada en una placa principal con un sencillo microcontrolador. Lo único que podemos decir como desventaja son los espacios que ocupan las placas de cada accesorio al momento de armar un proyecto.

Con arduino se pueden crear objetos interactivos, revisando los datos de una gran variedad de sensores y controlar todo tipo de luces, motores y otros actuadores físicos. La alimentación es a través del cable USB o con una fuente de alimentación externa de 12v. Como se muestra en la figura 13.



Figura 13. Arduino conexión
(Arduino CC)

1.6.15. GPS

El GPS es un sistema de radionavegación que se constituye por una constelación de 24 satélites que se encuentran en órbita, que proporciona servicios de posicionamiento, navegación, y cronometría totalmente gratuita e ininterrumpidamente a usuarios en todo el mundo. A cualquier persona que cuente con un receptor del GPS, el sistema le proporcionará su localización en longitud, latitud y altura a la hora exacta en cualquier tipo de condiciones atmosféricas, de día o de noche, en cualquier lugar del mundo y sin límite al número de usuarios, como se muestra en la figura 14.

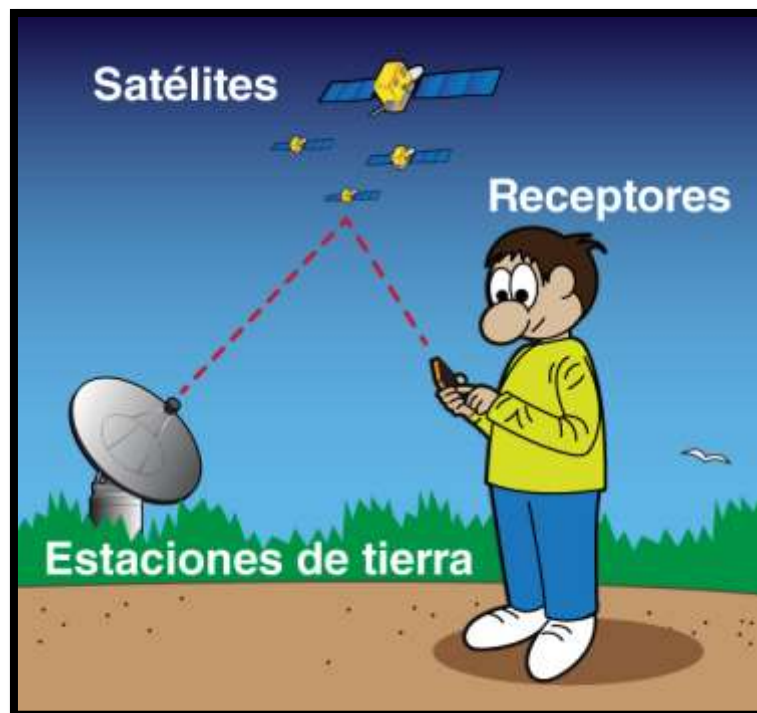


Figura 14. GPS
(Space place NASA)

El GPS se compone de tres elementos: los satélites que se encuentran en órbita, las estaciones terrestres de seguimiento y control, y los receptores del GPS que tenemos en cualquiera de nuestros dispositivos electrónicos. Estas señales contienen un código de tiempo y un punto de datos geográficos que permite al usuario identificar su posición exacta, la velocidad y el tiempo en cualquier parte de la tierra.

El GPS fue diseñado cuando estaban en la Guerra Fría en los años 60, su uso era específicamente para aplicaciones militares y del servicio de inteligencia. Luego fue creado para las aplicaciones civiles, control de espacios aéreos y terrestres para determinar su posición exacta, evitando errores de navegación y entradas involuntarias a los límites fronterizos. Como se muestra en la figura 15.



Figura 15. Navegación GPS
(Space place NASA)

CAPÍTULO II

ELEMENTOS DEL MODULO ARDUINO

2.1 Componentes y datos técnicos

Arduino uno es una placa de microcontrolador basada en la marca ATmega328P que significa microcontrolador de 8 bits (dígito binario), como se muestra en la figura 16. Tiene 14 pines digitales de entrada / salida, 6 entradas analógicas, un cristal de cuarzo de 16 MHz (megahercio), una conexión USB (bus serie universal), un conector de alimentación, un encabezado ICSP (programación serie de circuitos) y un botón de reinicio. Como se muestra en la tabla 1.

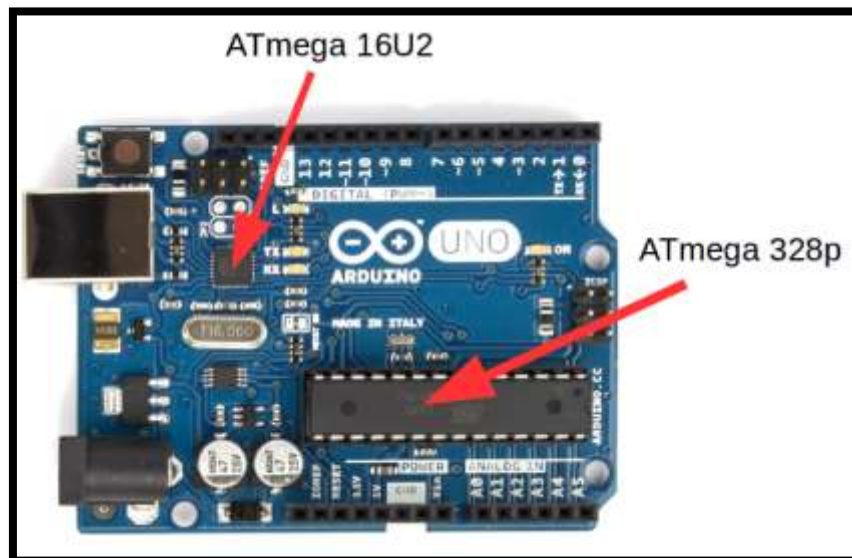


Figura 16. Microcontrolador ATmega328p
(Electronico Caldas)

Contiene todo lo necesario para soportar el microcontrolador; simplemente conéctelo a una computadora con un cable USB o con un adaptador de CA (corriente alterna) a CC (corriente continua) o una batería para comenzar.

"Uno" significa uno en italiano y fue elegido para marcar el lanzamiento del software Arduino versión 1.0. La placa Uno y la versión 1.0 de Arduino fueron las versiones de referencia de Arduino, ahora evolucionadas a versiones más recientes. El tablero Uno es el primero de una serie de placas USB Arduino, y el modelo de referencia para la plataforma Arduino. Como se muestra en la figura 17.



Figura 17. Arduino UNO
(Exploring Arduino)

Tabla 1. Elementos del Módulo Arduino (Exploring Arduino)

Componentes	Valores
Microcontrolador	ATmega328P
Voltaje de funcionamiento	5V
Voltaje de entrada	7-12V
Voltaje de entrada	6-20V
Pines de E / S	Digitales 14
Pasadores de E / S	Digitales PWM 6
Pasadores de entrada	Analógica 6
Corriente continua por perno E/S	20 mA
Corriente DC para 3.3V Pin	50 mA
Memoria Flash	32 KB (ATmega328P)
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Velocidad del reloj	16 MHz
LED integrado	13
Longitud	68.6 mm
Ancho	53.4 mm
Peso	25g

2.2. Pines de alimentación

El arduino se alimenta mediante un cable USB o mediante una fuente externa de 12v, se va a tener unas salidas de tensión continua debido a unos reguladores de tensión y condensadores de estabilización. Como se muestra en la tabla 2 y figura 18.

Tabla 2. Pines de alimentación (Exploring Arduino)

PINES	DESCRIPCIÓN
VIN	Se trata de la fuente tensión de entrada que contendrá la tensión a la que estamos alimentando al Arduino mediante la fuente externa.
5V	Fuente de tensión regulada de 5V, esta tensión puede venir ya sea de pin VIN a través de un regulador interno, o se suministra a través de USB o de otra fuente de 5V regulada.
3.3 V	Fuente de 3.3 voltios generados por el regulador interno con un consumo máximo de corriente de 50mA.
GND	Pines de tierra.

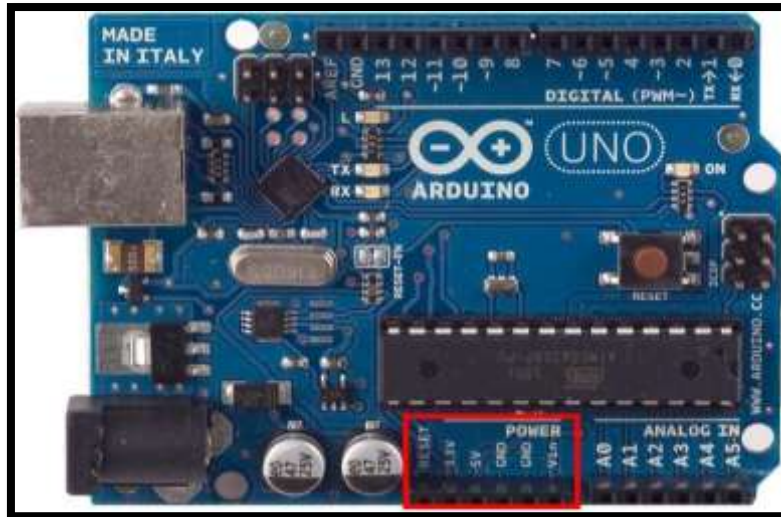


Figura 18. Pines de alimentación
(Exploring Arduino)

2.3. Entradas y salidas digitales

En la figura 19. Cada uno de los pines digitales pueden ser utilizados como entradas o salidas. Cada pin proporciona o recibe un máximo de 40 mA y tiene una resistencia de 20 a 50 kOhm. Además, algunos pines tienen funciones especializadas como se muestra en la tabla 3.

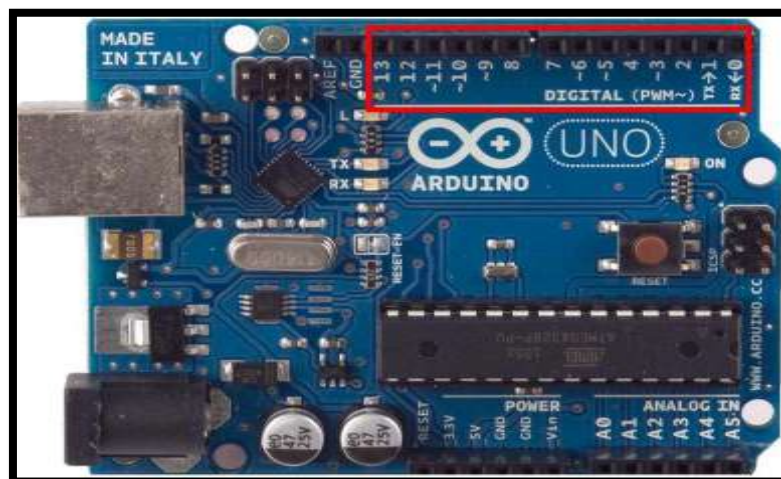


Figura 19. Pines de entrada y salida
(Explorando Arduino)

Tabla 3. Entradas y salidas digitales (Exploring Arduino)

PINES	DESCRIPCIÓN
Pin 0 (RX) y 1 (TX)	Se utiliza para recibir (RX) y la transmisión (TX) de datos serie TTL.
Pin 2 y 3	Interrupciones externas. Se trata de pines encargados de interrumpir el programa secuencial establecido por el usuario.
Pin 3, 5, 6, 9, 10 y 11	PWM (modulación por ancho de pulso)
Pin 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK)	Estos pines son de apoyo a la comunicación SPI.
Pin 13. LED.	Cuando el pin es de alto valor, el LED está encendido, cuando el valor está bajo, es apagado.

2.4. Entradas analógicas

El arduino posee 6 entradas analógicas, etiquetadas desde la A0 a A5, como se muestra en la figura 20, cada una de las cuales ofrecen 10 bits de resolución. Las placas del arduino tienen una tensión de 5V, pero podemos cambiar este rango utilizando el pin de AREF (referencia analógica) donde le introducimos una señal externa de continua que la utilizará como referencia.

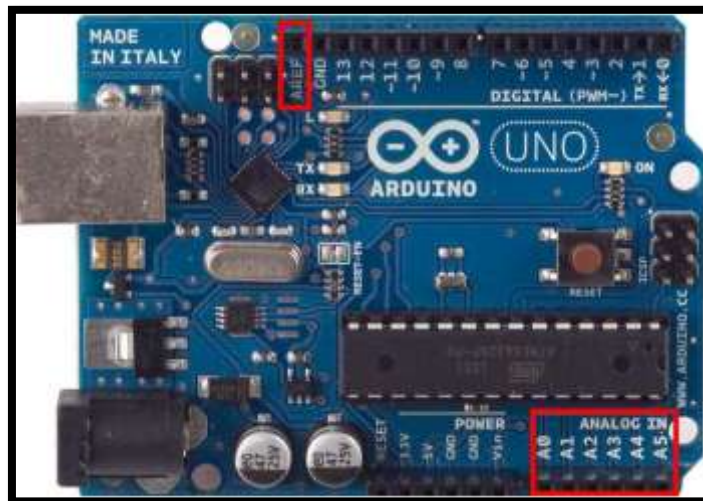


Figura 20. Entradas analógicas (Exploring Arduino)

2.5. Las salidas analógicas

Las siglas PWM (Modulación de Ancho de Pulso). Nos sirven para variar progresivamente la velocidad de un motor por medio de la frecuencia de la intensidad con un LED. Las salidas analógicas utilizan todavía algunos pines-hembra de salida digitales para “simular” el comportamiento analógico.

Estas señales lo que hace es emitir en lugar de una señal continua, una señal cuadrada formada por pulsos de frecuencia constante. Con esto variamos la duración de estos pulsos y su tensión.

Si los pulsos son más cortos, la tensión promedio de salida será menor, y cuanto más largos sean los pulsos, mayor será dicha tensión. La duración del pulso se puede cambiar en cualquier momento mientras la señal se está emitiendo. Como se muestra el diagrama 5.

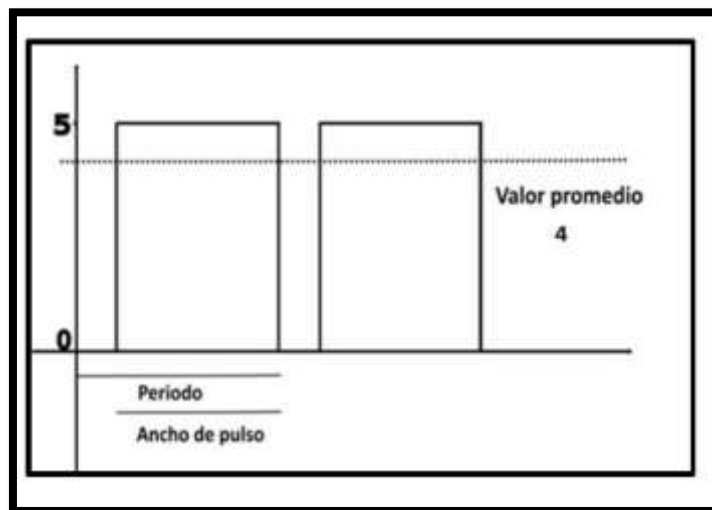


Diagrama 5. Salidas analógicas
(Ticarte)

2.6. Software Arduino

Arduino trabaja con un software gratis como se muestra en la figura 21, que debemos instalar en nuestro computador y que nos permite escribir, verificar y guardar en la memoria del microcontrolador de la placa Arduino la programación con el conjunto de instrucciones que deseamos que este empiece a ejecutar. Este software nos permite programarlo conectándolo a nuestro computador con la placa Arduino para poder enviarle y grabarle dichas instrucciones es mediante un simple cable USB.



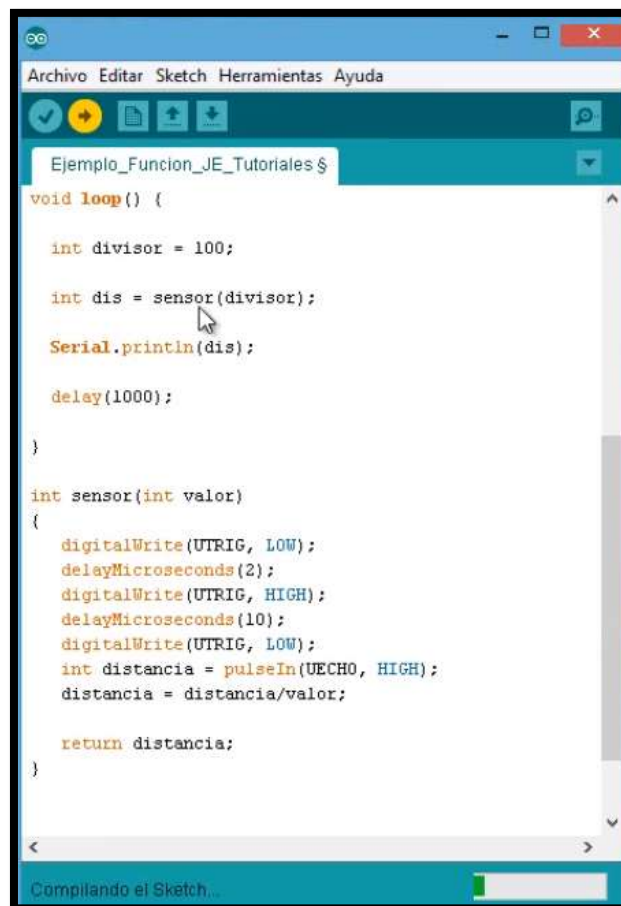
Figura 21. Software Arduino
(Ticarte)

Los proyectos con arduino, una vez programado su microcontrolador, la placa no necesita estar conectada a ningún computador y puede funcionar autónomamente si dispone de alguna fuente de alimentación.

La placa debe estar conectada ya sea por USB o Cable de red a un computador ejecutando el software específico que permita la comunicación entre este y la placa y el intercambio de datos entre ambos dispositivos.

Este software específico lo deberemos programar generalmente nosotros mismos mediante el lenguaje de programación C++ y será independiente completamente del entorno de desarrollo Arduino, el cual no se necesitará más, una vez que la placa ya haya sido programada y esté en funcionamiento.

En la figura 22 se puede apreciar el software arduino que lleva incorporado, la cual se necesita para elaborar la programación de cada componente que lleva.

The image shows a screenshot of the Arduino IDE interface. The window title is "Ejemplo_Funcion_JE_Tutoriales \$". The menu bar includes "Archivo", "Editar", "Sketch", "Herramientas", and "Ayuda". The toolbar contains icons for saving, uploading, and other functions. The main text area displays the following C++ code:

```
void loop() {  
  
  int divisor = 100;  
  
  int dis = sensor(divisor);  
  Serial.println(dis);  
  
  delay(1000);  
  
}  
  
int sensor(int valor)  
{  
  digitalWrite(UTRIG, LOW);  
  delayMicroseconds(2);  
  digitalWrite(UTRIG, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(UTRIG, LOW);  
  int distancia = pulseIn(UECHO, HIGH);  
  distancia = distancia/valor;  
  
  return distancia;  
}
```

At the bottom of the window, a status bar indicates "Compilando el Sketch..." with a progress bar.

Figura 22. Programación (Arduino CC)

Es un lenguaje de programación se entiende en cualquier idioma diseñado para expresar instrucciones lógicas que pueden ser llevadas a cabo por máquinas. Concretamente dentro del lenguaje Arduino, encontramos elementos parecidos a muchos otros lenguajes de programación existentes, así como también diferentes comandos que nos permiten especificar de una forma coherente y sin errores las instrucciones exactas que queremos programar en el microcontrolador de la placa.

2.7. El reloj o metrónomo

Para marcar el ritmo de ejecución de las instrucciones en el microcontrolador, la placa arduino posee un pequeño reloj, el cual funciona a una frecuencia de 16 millones de hercios (16MHz). Como se muestra en la figura 23.

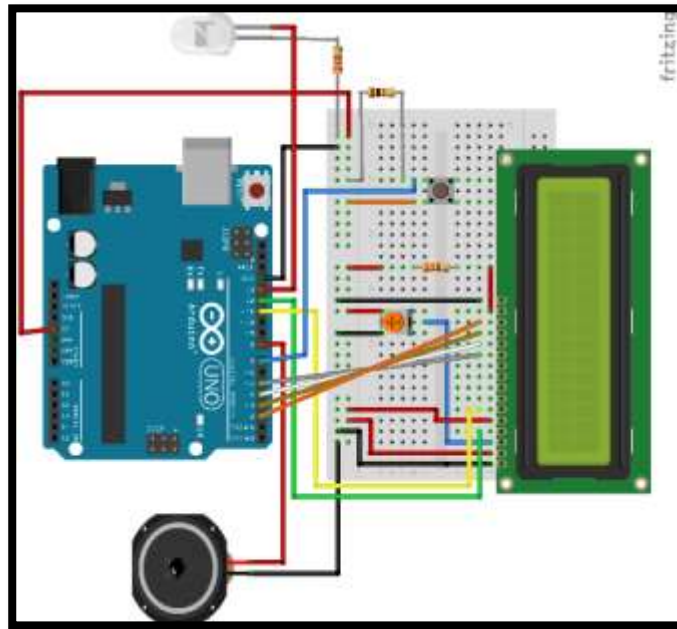


Figura 23. Metrónomo
(Arduino CC)

2.8. El botón de reinicio

La placa Arduino UNO dispone de un botón de reinicio (“reset”) como se muestra en la figura 24 que permite, una vez pulsado, enviar una señal LOW (Bajo) al pin “RESET” de la placa para parar y volver a arrancar el microcontrolador. Como en el momento del arranque del microcontrolador siempre se activa el modo ejecución, el botón de reinicio se suele utilizar para permitir la carga de un nuevo programa en la memoria Flash del microcontrolador –eliminando el que estuviera grabado anteriormente– y su posterior puesta en marcha.

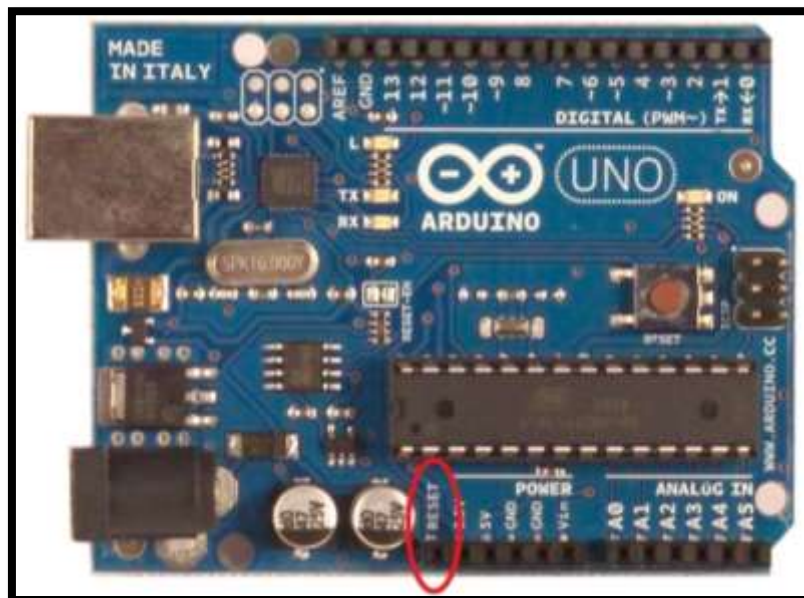


Figura 24. Botón reset
(Arduino CC)

CAPÍTULO III

HARDWARE Y SOFTWARE PARA EL DISEÑO Y CONSTRUCCIÓN DE ARDUINO Y GPS

3.1. Diseño del módulo

En este capítulo se desarrollará el diseño del módulo medidor de velocidad y aceleración del vehículo por Arduino y GPS, para ello se utilizan cinco etapas establecidos en selección de elementos, diagrama y conexiones, desarrollo interfaz LabView, programación en Arduino y pruebas de funcionamiento.

3.2. Implementación de hardware y software de Arduino.

La conexión a nuestro dispositivo Arduino se realiza a través de una placa con pantalla LCD DE 3,2" , dispositivo GPS o a su vez enlasearse con el telefono movil. La pantalla LCD solo nos servirá para revisar su funcionamiento después de haberlo programado. La conexión consta de 4 cables, 2 de señales(SCL-SDA) que van conectadas a los pines de la placa A4 y A5, tierra (GND) y voltaje continuo (VCC). Como se muestra en la figura 25.

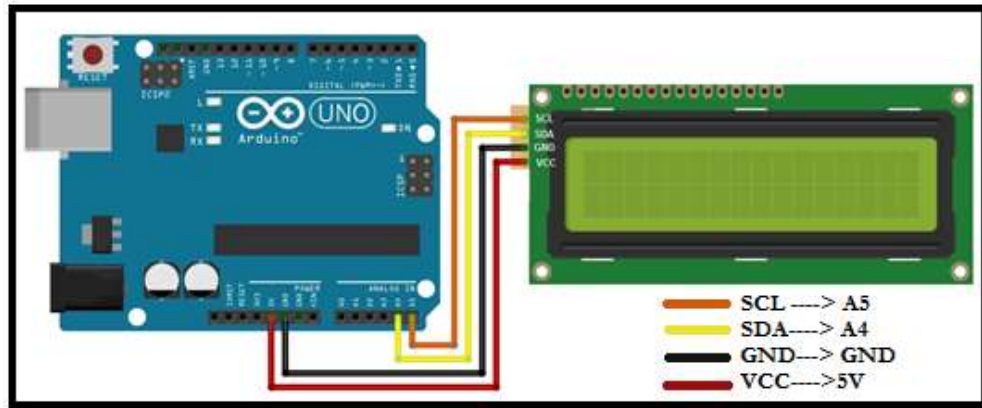


Figura 25. Conexión pantalla LCD (Arduino CC)

3.3. Construcción del Arduino

Para comenzar a contruir nuestro proyecto, todo lo que necesitará es un Arduino Uno, un receptor GPS con su placa de bajo, un cable de conexión corto y un receptor GPS EM -406A de 20 canales con antena incorporada como se muestra en la figura 26.

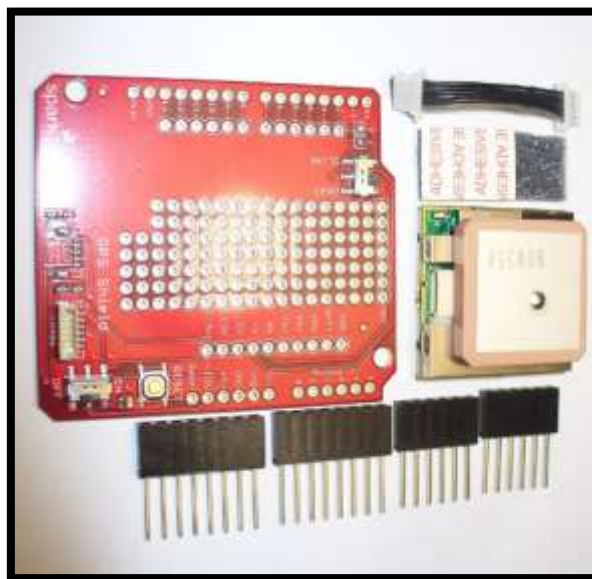


Figura 26. Placa GPS (Arduino CC)

Una vez que unimos las 2 placas tanto del Arduino como el del GPS nos debería de quedar así la conexión, tenga en cuenta que en la figura 27, el cable es mucho más largo entre la placa y el receptor GPS.

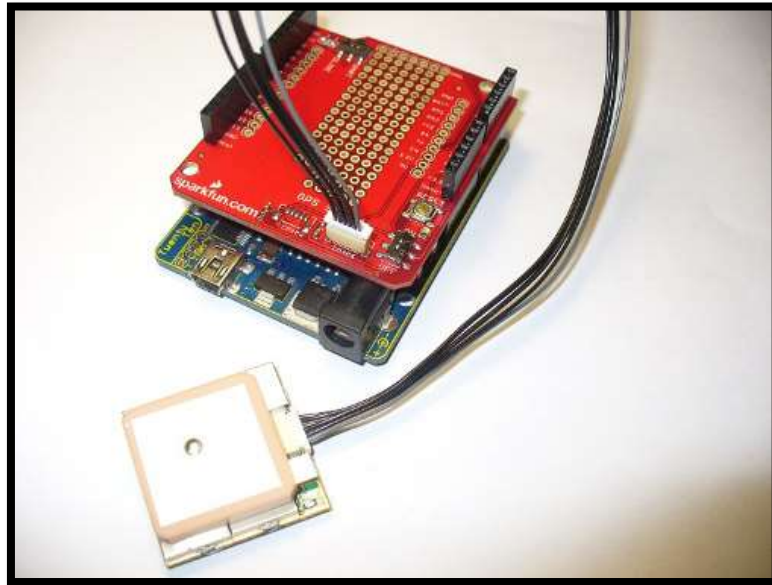


Figura 27. Conexión al módulo GPS
(Arduino CC)

En la figura 28 podemos apreciar con una imagen mas clara como nos quedarían la conexiones de la placa arduino al modulo GPS. Tenemos 4 cables, 2 de señal de color verde y amarillo, 1 cable de tierra de color negro y 1 cable de voltaje de color rojo.

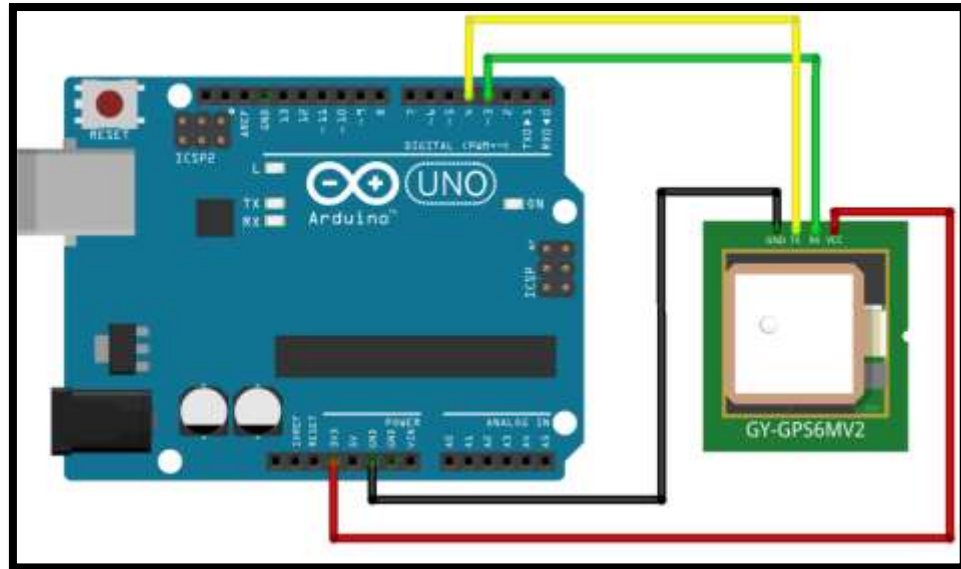


Figura 28. Cableado al módulo GPS (Arduino CC)

La placa arduino lleva un interruptor de encendido cerca del microcontrolador. Como se muestra en la figura 29.

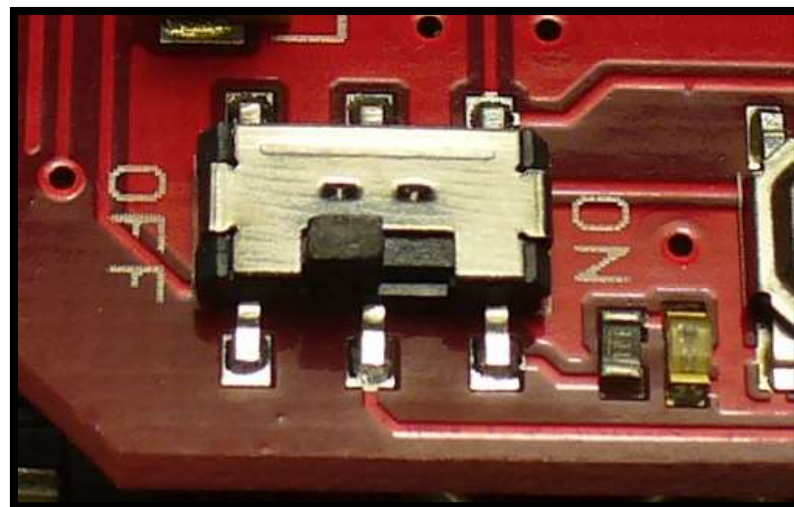


Figura 29. Interruptor GPS (Arduino CC)

3.4. Instalación del GPS

En la figura 30. Colocamos el GPS en nuestra cajita del Modulo Control de Velocidad (MCV) y tenemos que ubicar la antena en donde reciba señales de los satélites, el receptor deberá estar a la vista con el cielo. Una vez que encienda el modulo, el LED del receptor de GPS comienza a parpadear, esto significa que esta conectandose a los satelites disponibles, esto toma aproximadamente un minuto.



Figura 30. Señal GPS
(Kevin de la A.)

Esto estará con una línea de visión clara desde la unidad hacia el cielo. Una vez hecho esto, la próxima vez que lo encienda, el tiempo de búsqueda se reducirá un poco, ya que nuestro receptor almacena cierta energía en un supercap (condensador de muy alto valor) para recordar los datos del satélite,

que utilizará la próxima vez para reducir el tiempo de búsqueda. Ahora en el monitor en serie, se presentara unos codigos la cual nos va a indicar que esta funcinando correctamente. Los codigos que vamos a recibir se llaman NMEA (La Asociación Nacional de Electrónica Marina) en la siguiente tabla 4 demostrare los significados de cada codigo.

Tabla 4. Códigos del GPS (Exploring Arduino)

Códigos	Significado
\$ GPRMC	Nos dice que los siguientes datos son datos esenciales de punto-velocidad-tiempo
165307,000	Es la constante de tiempo universal 16:53:07 (horas, minutos, segundos). Entonces ahora también tienes un reloj.
A	A para activo y datos válidos, V para vacío y datos no válidos.
2728.9620	Es grados datos de posición de latitud = 27 grados, 28.962 ' S para el sur (el sur es negativo, el norte es positivo)
15259.5159	Es datos de posición de longitud de grados = 152 grados, 59.5159 ' E para este (este es positivo, oeste es negativo)
0.20	Es mi velocidad en nudos sobre el suelo. Esto muestra la inexactitud que puede ser causada por no tener una vista clara del cielo
48.84	Curso sobre el suelo (0 es norte, 180 es sur, 270 es oeste, 90 es este)
140118	Es la fecha - 14 de enero de 2018

El archivo es un archivo de texto típico, delimitado por comas. Puede examinarlo aún más utilizando un software de hoja de cálculo común, como LibreOffice Calc. Por ejemplo, si abre el archivo de datos GPS desde aquí, como se muestra en la figura 31.



Figura 31. Archivo de datos
(Matrix TCL)

Puede ver que los datos se delimitan con bastante facilidad. Simplemente haga clic en "Aceptar" y se le presentará el archivo

Como puede ver en la figura 32 hay tiempo, fecha , latitud y longitud, velocidad y resultados de datos de sensor aleatorios. Podemos hacer que estos datos se conviertan en una forma mucho más útil mediante el uso del sitio web GPS Visualiser. Guarde los datos como un archivo .csv. Luego, visite <http://www.gpsvisualizer.com/> y use el cuadro Comenzar ahora en el medio de la página web. Seleccione Google Maps como formato de salida y luego cargue el archivo.

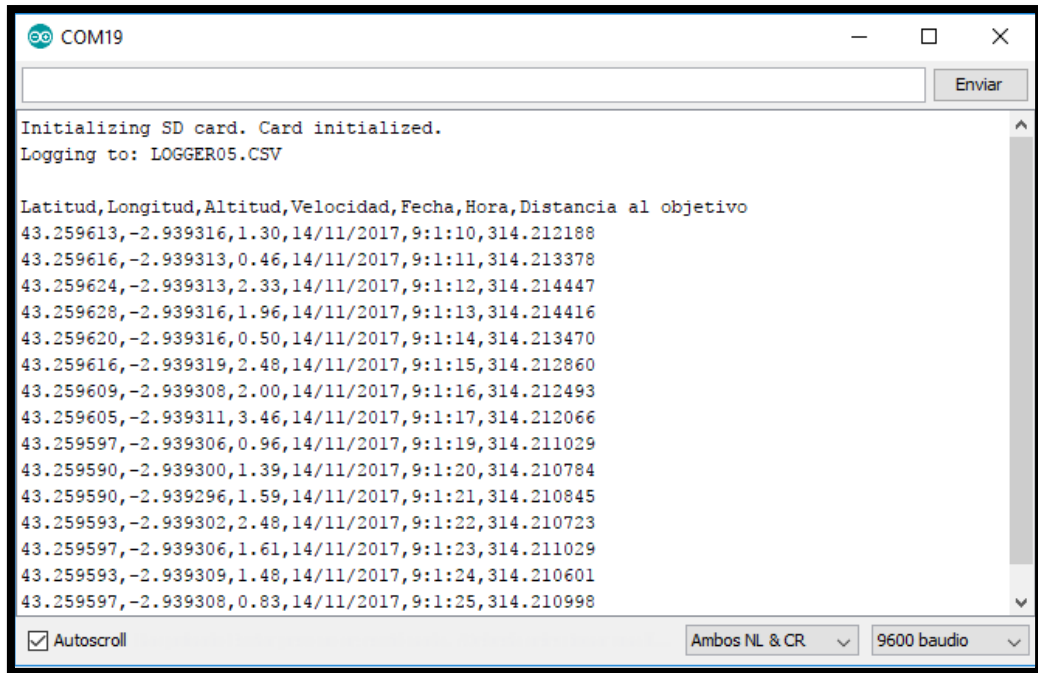


Figura 32. Ingreso de datos GPS
(Matrix TCL)

También podemos ingresar la información del módulo GPS en nuestra aplicación en la web de Google Maps e ingresamos las coordenadas de longitud y latitud. Como se muestra en la figura 33.



Figura 33. Ingreso Google Maps
(Google Maps)

3.5. Visualizacion de datos GPS en Arduino

En nuestro software Arduino podemos apreciar como registra los datos de latitud, longitud, velocidad, hora y fecha mediante una tarjeta SD (Seguridad Digital) que llevamos instalada en nuestro sistema. Como se muestra en la figura 34.

```
COM34  
Lat: -37.91662 Long: 145.11712  
GMT - 10:26:35 14/1/2014  
Altitude 92.50m 0.00km/h  
-----  
Lat: -37.91662 Long: 145.11712  
GMT - 10:26:35 14/1/2014  
Altitude 92.50m 0.00km/h  
-----  
Lat: -37.91662 Long: 145.11712  
GMT - 10:26:36 14/1/2014  
Altitude 92.50m 0.00km/h  
-----  
Lat: -37.91662 Long: 145.11712  
GMT - 10:26:36 14/1/2014  
Altitude 92.50m 0.00km/h  
-----  
Lat: -37.91662 Long: 145.11712  
GMT - 10:26:37 14/1/2014  
Altitude 92.50m 0.00km/h  
-----  
Lat: -37.91662 Long: 145.11712  
GMT - 10:26:37 14/1/2014  
Altitude 92.50m 0.00km/h  
-----  
 Autoscroll
```

Figura 34. Ingreso datos Arduino (Arduino Inc)

3.6. Programación de hora

Se requiere un poco de trabajo adicional para mostrar su hora local. Primero necesitará saber en qué zona horaria se encuentra, para esto revisamos en internet la zona horaria de nuestro país y registramos en nuestro software Arduino.

Si su zona horaria es positiva (por ejemplo, GMT +10), necesita agregar 10 a su valor de hora, y si es mayor de 23, resta 24 para obtener las horas correctas. Si su zona horaria es negativa (por ejemplo, GMT - 5), necesita restar 5 del valor de su hora, y si es bajo cero, entonces agregue 24 para obtener las horas correctas.

Necesitaremos los siguientes componentes como se muestra en la tabla 6.

Tabla 5. Horario (Geek Factory)

Cantidad	Componentes
1	Arduino Uno
1	Paquete de datos de GPS
1	Maxim MAX7219 controlador de pantalla IC
2	Pantallas LED de cuatro segmentos y siete segmentos (cátodo común)
1	Resistencia de 1 kilo ohmio
1	Resistencia de 10 kilo ohmios
1	Interruptor de doble tiro un polo
1	Protoboard y cables de conexión
1	Fuente de alimentación de 5V separada

Para hacer primero las pruebas utilizamos el programa de Livewire para comprobar que no tengamos ningún error al momento de armar nuestro circuito. Como se muestra en el diagrama 6.

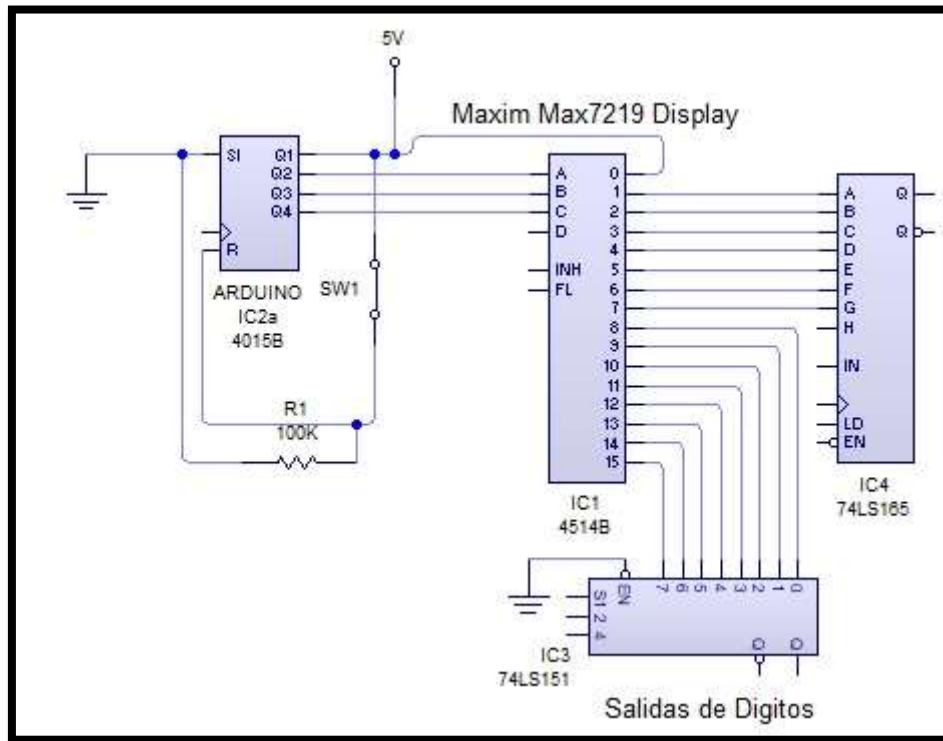


Diagrama 6. Horario
(Kevin de la A.)

Necesitará módulos de visualización de cátodo común. Si alinea los dígitos de izquierda a derecha, se numerarán de cero a nueve con respecto al MAX7219, así que conecte el pin 2 de MAX7219 al cátodo de su primera pantalla, y así sucesivamente. Como se muestra en la figura 35.

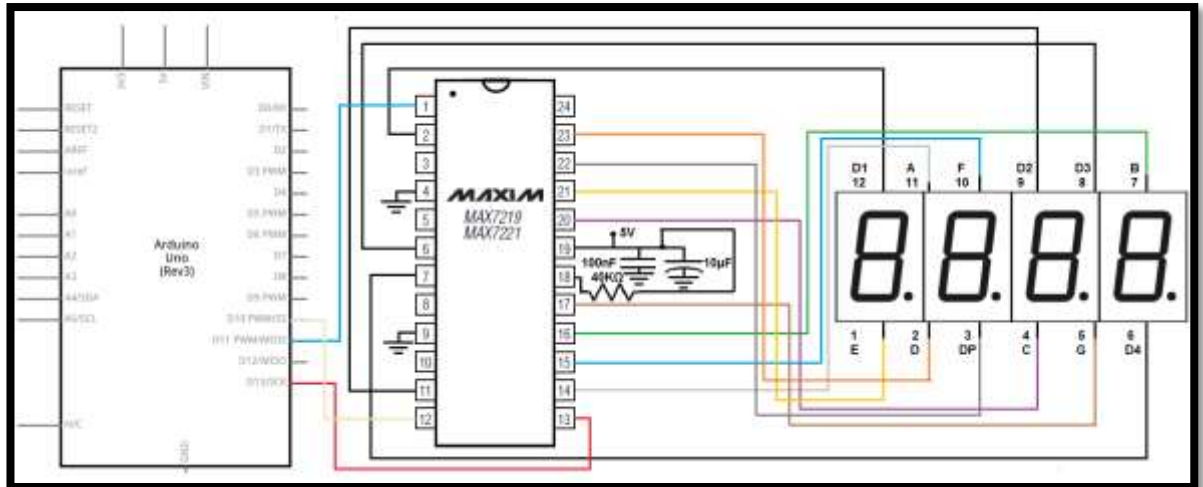


Figura 35. Conexiones de Max 7219
(Taller KR)

3.7. Velocímetro

Al igual que con el reloj, es fácil mostrar las lecturas de velocidad con la pantalla LCD. Usando el mismo hardware que el anterior.

Ahora coloque el hardware de nuevo, y después de un momento su velocidad debería aparecer. Puede obtener algunas lecturas extrañas si está en interiores, ya que el receptor necesita datos de varios satélites para determinar con precisión su velocidad. El boceto está escrito para km / h.

3.8. Tarjeta de memoria

Los datos de posición, velocidad y tiempo en la tarjeta de memoria SD para un análisis posterior (figura 36). El hardware requerido es solo la placa del GPS y Arduino Uno, más una tarjeta de memoria SD que está formateada en FAT16.

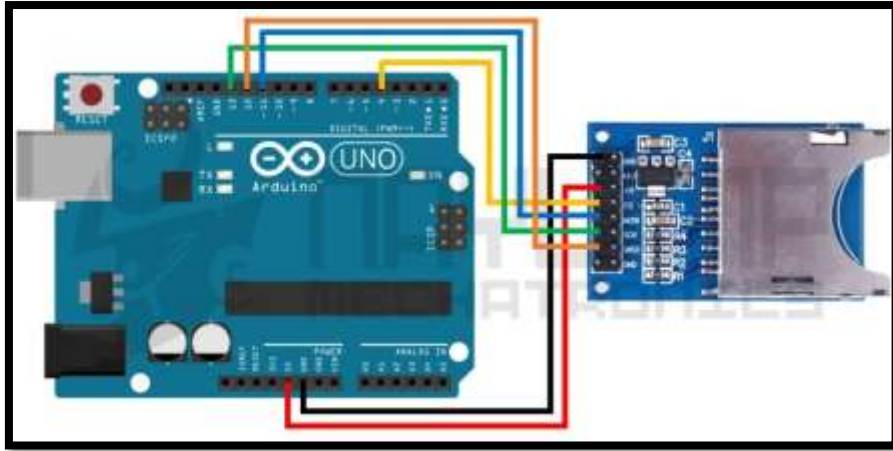


Figura 36. Conexión de tarjeta SD
(Arduino Inc.)

En la figura 37, podemos apreciar cómo nos queda instalada el módulo de la tarjeta SD en nuestra cajita de Modulo Control de Velocidad (MCV).

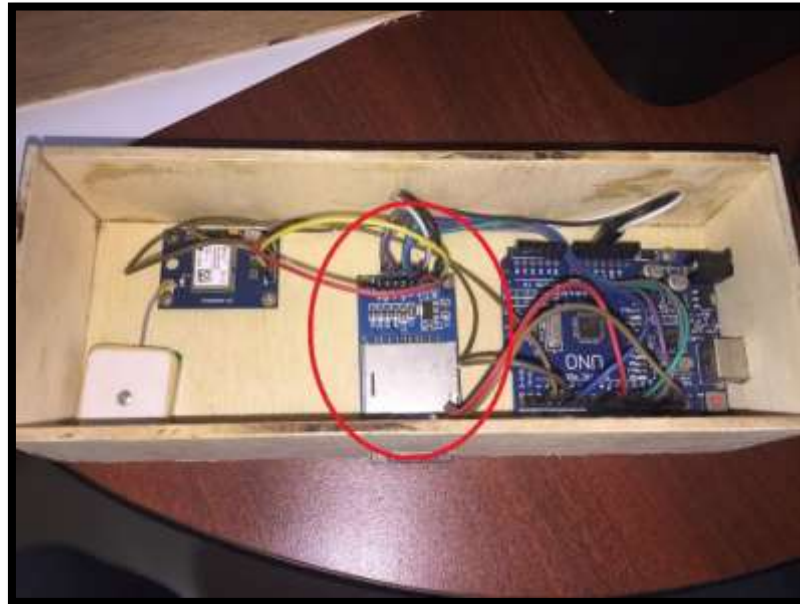


Figura 37. Tarjeta SD
(Kevin de la A.)

Esto agregará los datos a un archivo de texto. Los datos se pueden registrar cada diez segundos o como el usuario desea, se pueden cambiar los parámetros en la programación.

Cuando esté listo para comenzar a capturar datos, simplemente inserte la tarjeta y encienda el Arduino. Continuará hasta que lo apague, momento en el cual el archivo de datos se puede examinar en una computadora, en el programa de bloc de notas. Como una se muestra en la figura 38.

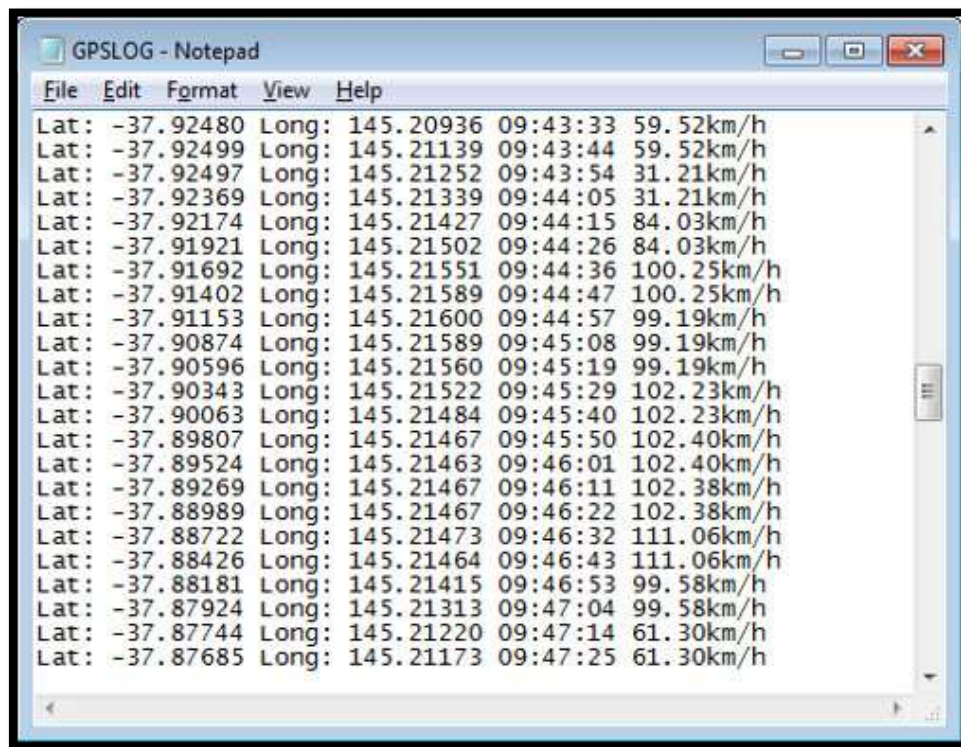


Figura 38. Registro velocidad
Fuente: (Arduino Inc.)

Ahora guarde ese archivo como una hoja de cálculo en excel y luego visite el sitio web de Visualizer de GPS y cargue el archivo. Seleccione "Google Maps" como formato de salida y se presentará como se muestra en la figura 39.

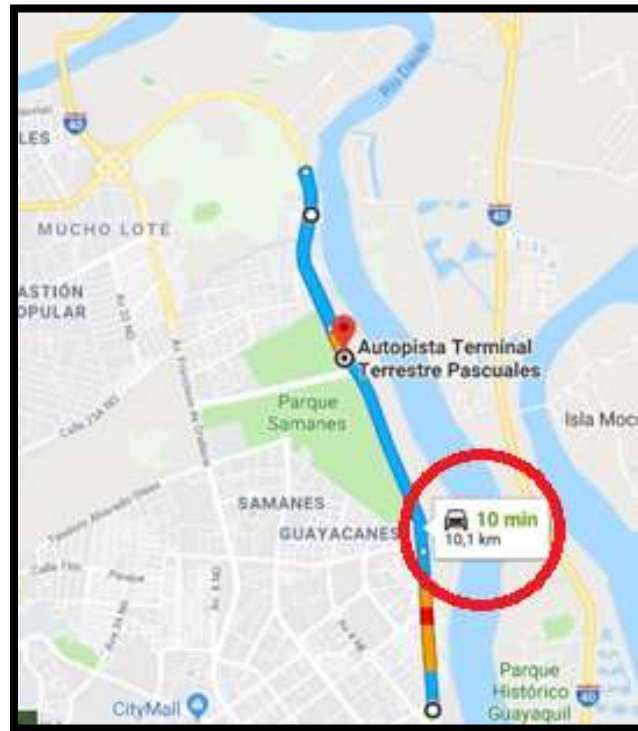


Figura 39. Monitoreo GPS
Fuente: (Google Maps)

CAPÍTULO IV

COMPROBACIÓN Y RESULTADOS DE UN MÓDULO MEDIDOR DE VELOCIDAD Y ACELERACIÓN DEL VEHÍCULO POR ARDUINO Y GPS.

4.1. Comprobación y funcionamiento del módulo

Vamos a comprobar el funcionamiento de nuestro módulo Arduino por medio de pruebas de velocidad y aceleración en nuestro vehículo, revisando que nos de como resultado los diferentes rangos de velocidad y tiempo de nuestro trayecto con la ubicación exacta. Vamos a necesitar un vehículo y alimentación de 12v (voltios).

4.2. Modo de emplearse

Montamos nuestro modulo MCV dentro de la cabina de un auto Volkswagen, y se ubico detrás de la guantera como se muestra en la figura 40.



Figura 40. Montaje en el vehículo.
(Kevin de la A.)

4.3. Instalación de antena GPS en el vehículo

Instalamos la antena GPS (Sistema de posicionamiento Global) en la esquina del parabrisas delantero del vehículo para obtener una óptima señal con los satélites. Para esto tenemos que remover el plástico frontal del panel delantero para pasar el cable de la antena. Como se muestra en la figura 41.



Figura 41. Instalación de antena GPS.
(Kevin de la A.)

Nos tiene que quedar fijo en el tablero y con visibilidad hacia el cielo. Como se muestra en la figura 42.



Figura 42. Montaje antena GPS.
(Kevin de la A.)

Luego conectamos la batería de 9v (Voltios) que se encuentra dentro de nuestra cajita MCV. Como se muestra en la figura 43



Figura 43. Alimentación 9v.
(Kevin de la A.)

Hacemos una prueba de ruta por la autopista Terminal-Pascuales, donde tenemos como limite de velocidad 90km/h (kilometros/ hora). Esta autopista consta de 7 radares de control de velocidad de la ATM.

4.4. Revision de datos en la memoria SD

Una vez que concluimos nuestra prueba de ruta, retiramos la memoria SD que se encuentra dentro de nuestro modulo Arduino y revisamos los datos de la prueba dentro de nuestra computadora, abrimos el archivo y lo copiamos toda la informacion al programa de bloc de notas para tenerlo ya en formato .txt. Como se muestra en la figura 44.

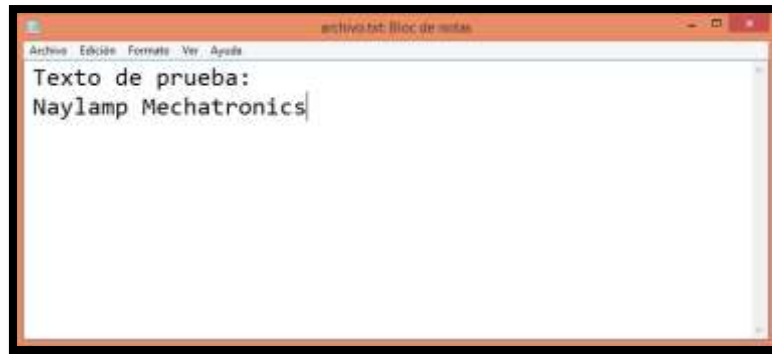


Figura 44. Archivo .txt.
(Bloc de notas)

4.5. Registro de datos en el programa

Luego copiamos todo en el archivo de texto bloc de notas, guardamos el archivo y lo abrimos con el programa o sistema arduino, para verificar la información por medio de graficos y rangos de velocidad en las coordenadas obtenidas por el GPS. Como se muestra en la figura 45.

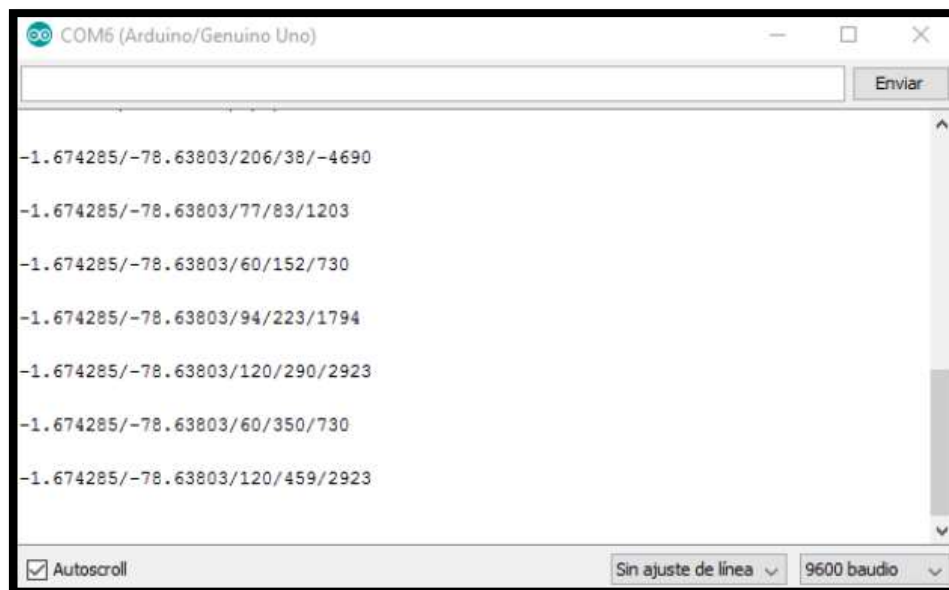


Figura 45. Registro de datos
(Arduino Inc.)

Finalmente estas coordenadas que nos da el modulo arduino son pasadas a la tabla de grafica del mismo programa para revisar los rangos de velocidad, como se muestra en la figura 46.

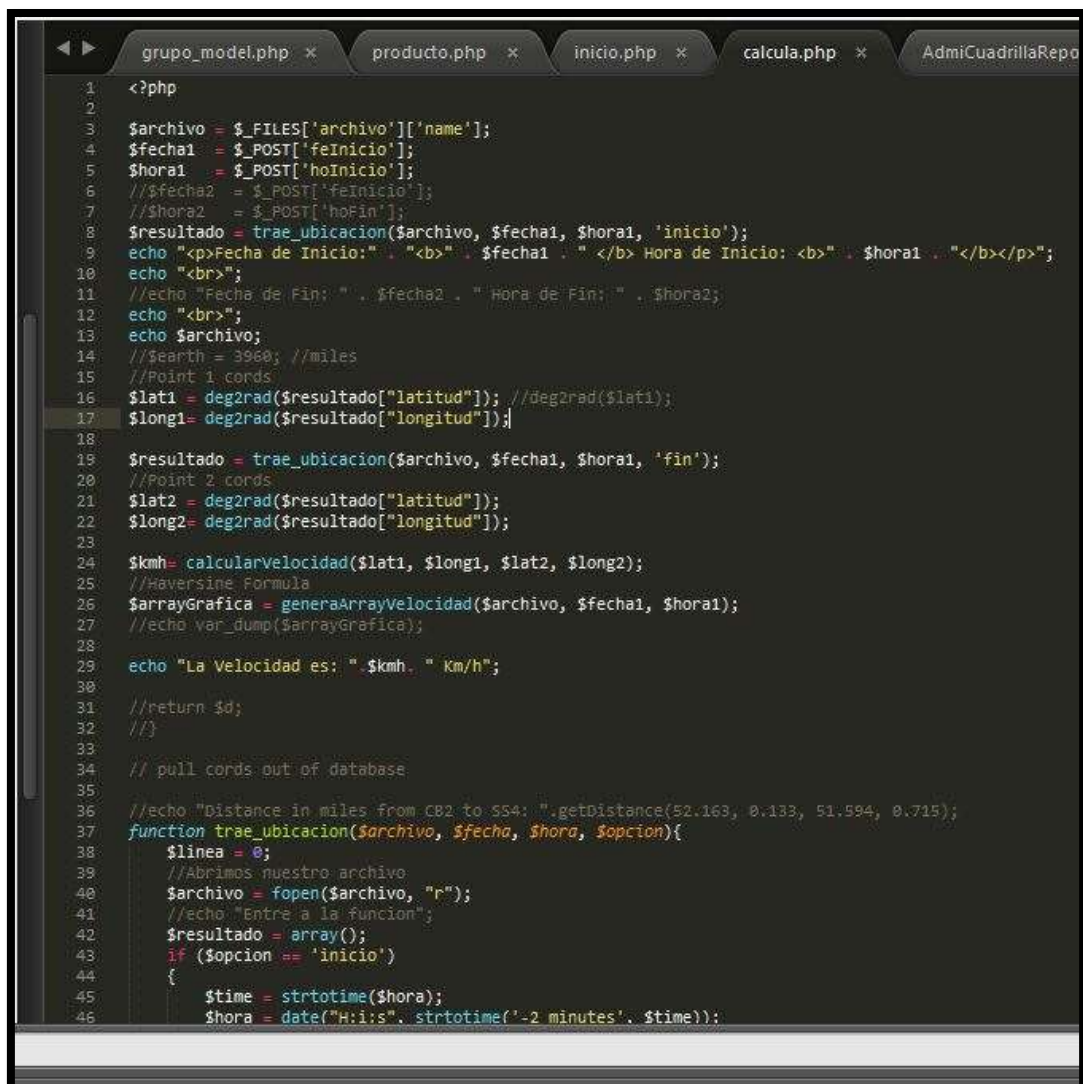


Figura 46. Ubicación GPS
(Arduino Inc.)

4.6. Programación

Basicamente la programación del arduino es para registrar cada movimiento que se desea hacer, con sus diferentes tipos de complementos que llevan.

No es fácil la codificación, ya que para poder realizar esta programación se necesita tener conocimientos del lenguaje C++. Para poder programar un arduino es mejor recurrir a un programador que nos pueda ayudar a codificar para que pueda trabajar correctamente y así obtener óptimos resultados. En la figura 47 se muestra la programación de nuestro arduino.



```
1 <?php
2
3 $archivo = $_FILES['archivo']['name'];
4 $fecha1 = $_POST['feInicio'];
5 $hora1 = $_POST['hoInicio'];
6 // $fecha2 = $_POST['feFin'];
7 // $hora2 = $_POST['hoFin'];
8 $resultado = trae_ubicacion($archivo, $fecha1, $hora1, 'inicio');
9 echo "<p>Fecha de Inicio: " . "<b>" . $fecha1 . " </b> Hora de Inicio: <b>" . $hora1 . "</b></p>";
10 echo "<br>";
11 //echo "Fecha de Fin: " . $fecha2 . " Hora de Fin: " . $hora2;
12 echo "<br>";
13 echo $archivo;
14 // $earth = 3960; //miles
15 // Point 1 coords
16 $lat1 = deg2rad($resultado["latitud"]); //deg2rad($lat1);
17 $long1 = deg2rad($resultado["longitud"]);
18
19 $resultado = trae_ubicacion($archivo, $fecha1, $hora1, 'fin');
20 // Point 2 coords
21 $lat2 = deg2rad($resultado["latitud"]);
22 $long2 = deg2rad($resultado["longitud"]);
23
24 $kmh = calcularVelocidad($lat1, $long1, $lat2, $long2);
25 // Haversine Formula
26 $arrayGrafica = generaArrayVelocidad($archivo, $fecha1, $hora1);
27 //echo var_dump($arrayGrafica);
28
29 echo "La Velocidad es: " . $kmh . " Km/h";
30
31 //return $d;
32 //}
33
34 // pull coords out of database
35
36 //echo "Distance in miles from CB2 to SS4: ".getDistance(52.163, 0.133, 51.994, 0.715);
37 function trae_ubicacion($archivo, $fecha, $hora, $opcion){
38     $linea = 0;
39     //Abrimos nuestro archivo
40     $archivo = fopen($archivo, "r");
41     //echo "Entré a la función";
42     $resultado = array();
43     if ($opcion == 'inicio')
44     {
45         $time = strtotime($hora);
46         $hora = date("H:i:s", strtotime('-2 minutes', $time));
```

Figura 47. Programación
(Kevin de la A.)

La programación se creó en base al trabajo que se desea hacer en nuestro arduino, para registrar información de velocidad, aceleración y ubicación real del vehículo.

4.7. Curva gráfica de prueba

En la figura 48 tenemos los resultados de la prueba utilizando el programa arduino. Podemos observar que nuestro vehículo no iba a una velocidad constante, esto se debe a que iba en una zona donde el límite de velocidad permitido es de 25km/h.

En la gráfica podemos ver el tiempo y la hora de inicio del programa al momento de prender el arduino.

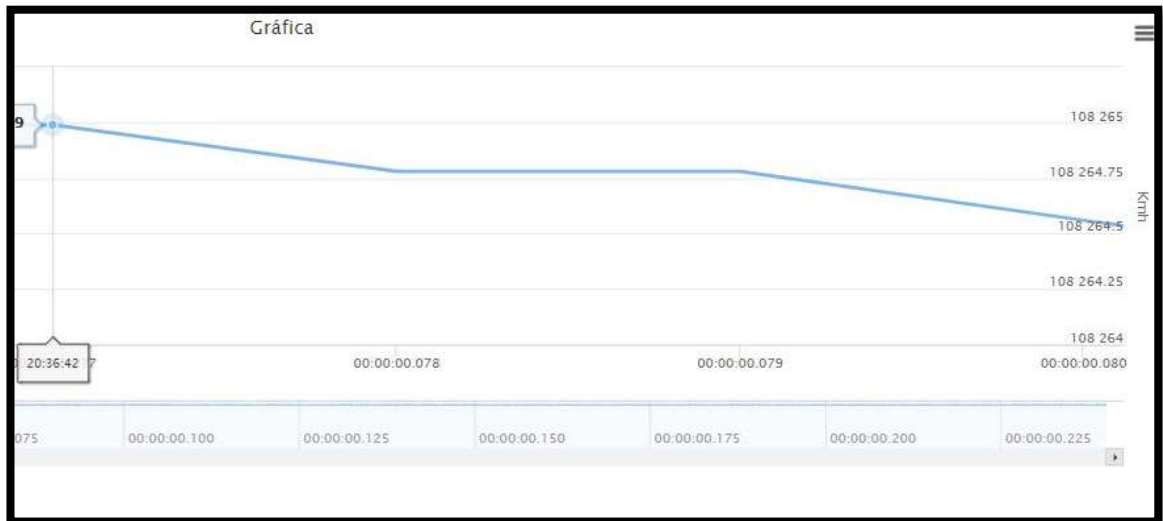


Figura 48. Curva gráfica de la velocidad del vehículo (Kevin de la A.)

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- Al realizar pruebas con este proyecto por medio de arduino se adquirió el conocimiento suficiente de su funcionamiento técnico y eficiente ya que registra en la memoria valores en tiempo real del módulo medidor de velocidad y aceleración.
- Al analizar los resultados teóricos y prácticos de este módulo, proporcionará información precisa para comprobar los rangos de velocidad con la que va el vehículo, la cual beneficiará al usuario con los registros de velocidad y localización precisas al momento de tener una infracción que no le corresponda.
- Al investigar la información sobre las funciones de nuestro módulo pudimos aprender que podemos trabajar en diferentes proyectos en la cual nos beneficiaría para registrar cualquier tipo de información de nuestro vehículo en tiempo real.

5.2. Recomendaciones

- Al realizar las conexiones del módulo arduino utilizar siempre las herramientas adecuadas y seguir los pasos indicados en el manual del usuario ya que podemos generar un corto en el sistema y dañar algún componente interno.
- Se recomienda revisar bien la programación en el programa de arduino ya que, si falta un punto o una coma, no va a trabajar correctamente o vamos a tener resultados erróneos.
- Siempre apagar el módulo arduino desde el interruptor principal, si lo desconecta directamente desde la fuente de alimentación, podemos perder información real que está siendo registrada en el momento en la tarjeta de memoria.

BIBLIOGRAFÍA

- Almendros, J. J. (2016). *Programacion visual con bloques Arduino*. Create Space independent .
- Aranda, D. (2014). *Electronica: Plataformas arduino y rasperry PI*. Argentina: Fox andina.
- Arduino. (2018). *Arduino CC*. Obtenido de www.arduino.cc
- Bayle, J. (2013). *Programacion C para arduino*. UK: Packt Publishing Ltd.
- Benzi, M. (2012). *Introduccion a Arduino*. Anaya multimedia.
- Corona, L. G. (2014). *Sensores y actuadores aplicaciones con arduino*. Mexico D.F.: Grupo Patria.
- Crespo, E. (s.f.). *Aprendiendo Arduino*. Obtenido de www.aprendiendoarduino.wordpress.com
- Goilav, N. (2016). *Aprender a desarrollar para crear objetos inteligentes*. Barcelona: ENI.
- Lequerica, J. R. (2014). *Arduino Practico*. Anaya multimedia.
- Predrera, A. C. (2017). *Arduino para principiantes*. IT.
- Usategui, J. M. (2007). *Electronica digital y microprogramable*. Madrid: Paraninfo.
- W, R. (2008). *Electromecanica de vehiculos*. Mexico D.F.: Limusa.

ANEXOS

GLOSARIO

ABREVIATURAS	SIGNIFICADO
GPS	SISTEMA DE POSICIONAMIENTO GLOBAL
ATM	AUTORIDAD DE TRANSITO MUNICIPAL
mV	MILIVOLTIOSS
V	VOLTIO
kV	KILOVOLTIO
A	AMPERIO
mA	MILIAMPERIOS
uA	MICRO AMPERIO
W	VATIO
mW	MILIVATIOS
nA	NANO AMPERIO
R	RESISTENCIA
I	INTENCIDAD
V	POTENCIA
LED	DIODO EMISOR DE LUZ