



INSTITUTO POLITECNICO NACIONAL

UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA Y TECNOLOGÍAS AVANZADAS

UPIITA

Proyecto Integrador

“Sistema de encendido biométrico y ajuste de espejos laterales”,

para obtener el título de

“Ingeniero en Sistemas Automotrices”

Presenta

Jair Humberto Espinoza Pulido

Asesores

Dra. Blanca Briseño Tepepa

Dr. José Alejandro Santa Cruz



Ciudad de México, a 27 de abril del 2022



INSTITUTO POLITECNICO NACIONAL

UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERIA Y TECNOLOGIAS AVANZADAS

“Sistema de encendido biométrico y ajuste de espejos laterales”

Proyecto integrador

Ingeniería en Sistemas Automotrices

Presenta:

Jair Humberto Espinoza Pulido

Dra. Blanca Briseño Tepepa
Asesora

Dr. José Alejandro Santa Cruz
Asesor

M. en C. Víctor Darío Cuervo Pinto
Secretario

Dra. Laura Ivoone Garay Jiménez
Presidenta



Autorización de uso de obra

Instituto Politécnico Nacional

P r e s e n t e

Bajo protesta de decir verdad el que suscribe Jair Humberto Espinoza Pulido (se anexa copia simple de identificación oficial), manifiesto ser autor (a) y titular de los derechos morales y patrimoniales de la obra titulada Sistema de Encendido Biometrico y Ajuste de Espejos Laterales, en adelante "La Tesis" y de la cual se adjunta copia, por lo que por medio del presente y con fundamento en el artículo 27 fracción II, inciso b) de la Ley Federal del Derecho de Autor, otorgo a el Instituto Politécnico Nacional, en adelante El IPN, autorización no exclusiva para comunicar y exhibir públicamente total o parcialmente en medios digitales, Plataforma de la Dirección de Bibliotecas del IPN y/o consulta directa en la Coordinación de Biblioteca de la UPIITA "La Tesis" por un periodo de 5 años contado a partir de la fecha de la presente autorización, dicho periodo se renovará automáticamente en caso de no dar aviso expreso a "El IPN" de su terminación. En virtud de lo anterior, "El IPN" deberá reconocer en todo momento mi calidad de autor de "La Tesis".

Adicionalmente, y en mi calidad de autor y titular de los derechos morales y patrimoniales de "La Tesis", manifiesto que la misma es original y que la presente autorización no contraviene ninguna otorgada por el suscrito respecto de "La Tesis", por lo que deslindo de toda responsabilidad a El IPN en caso de que el contenido de "La Tesis" o la autorización concedida afecte o viole derechos autorales, industriales, secretos industriales, convenios o contratos de confidencialidad o en general cualquier derecho de propiedad intelectual de terceros y asumo las consecuencias legales y económicas de cualquier demanda o reclamación que puedan derivarse del caso.

Ciudad de México. 27 de Abril de 2022.

Atentamente

1. Agradecimientos

Para llegar hasta donde estoy actualmente no es un trabajo que se logra por si solo, se necesita apoyo de la gente que te rodea, familiares, maestros y amigos, es por eso que quiero agradecer a todos los que me ayudaron a cumplir mis metas y poder terminar mi carrera satisfactoriamente.

Empezando por mis papás, ya que ellos me dieron las facilidades de poder estudiar en una escuela tan prestigiosa, desde lo económicamente y hasta lo anímicamente ellos fueron los principales que me impulsaron a lograr mis metas. En el trayecto de mi vida universitaria me encontré con maestros y amigos que me apoyaron con dudas, problemas y obstáculos que tuve con ciertos temas y principalmente durante al desarrollo de mi proyecto de titulación, ya que mis amigos me apoyaron con el automóvil donde iba a instalar mi sistema, a mis asesores que me apoyaron y me guiaron durante todo el desarrollo de mi proyecto.

Finalmente quiero agradecer a la institución que me dio todas las herramientas cognitivas para poder desarrollarme como profesional.

2. Resumen

El proyecto desarrollado tendrá la función de incrementar seguridad y confort al usuario que lo esté manejando, debido a que el sistema de encendido será mediante un sensor de huellas digitales y a la vez se ajustaran los espejos dependiendo de quién lo encienda.

3. Abstract

The project developed is intended to give more security and confort to the driver, the motor will turn on with a fingerprint sensor and the review mirrors will adjust to the driver's preferences.

Índice

1.	AGRADECIMIENTOS	2
2.	RESUMEN	3
3.	ABSTRACT	3
4.	ÍNDICE DE ILUSTRACIONES	5
5.	ÍNDICE TABLAS	6
6.	ÍNDICE DIAGRAMA DE FLUJO	6
7.	ÍNDICE LÍNEA DE CÓDIGO	6
8.	ÍNDICE ABREVIATURAS	7
9.	INTRODUCCIÓN	8
9.1	PLANTEAMIENTO DEL PROBLEMA	8
9.2	OBJETIVO GENERAL	8
9.3	OBJETIVOS PARTICULARES	9
9.4	JUSTIFICACIÓN	9
9.5	ANTECEDENTES	10
9.5.1	<i>Huella digital Seguridad automotriz Sistema (Rodanliu OSBS001)</i>	10
9.5.2	<i>Sistema de huella dactilar para encender motor de Hyundai</i>	12
10.	MARCO TEÓRICO	13
10.1	HUELLA DIGITAL	13
10.2	GIROSCOPIO	14
10.3	MICROCONTROLADOR	15
10.3.1	<i>Arquitectura interna</i>	16
10.3.2	<i>Arquitectura Von Neumann</i>	17
10.3.3	<i>Arquitectura Harvard</i>	17
10.4	DISPLAY LCD	18
10.5	PUENTE H	19
11.	DESARROLLO	21
11.1	DIAGRAMA METODOLOGÍA V	21
11.2	SENSOR HUELLA DIGITAL	22
11.3	SENSOR MPU6050	23
11.4	ARDUINO	24
11.5	DISPLAY LCD	27
11.6	PUENTE H LM293D	29
12.	RESULTADOS	31
12.1	FUNCIONAMIENTO GENERAL	31
12.2	CONTROL ESPEJOS	32
12.3	PUENTE H LM293D	33
12.4	SENSOR MPU6050	35
12.5	SENSOR DE HUELLA DIGITAL	41
12.6	DISPLAY LCD	45

12.7 ALIMENTACIÓN	48
12.8 INTEGRACIÓN DE TODOS LOS COMPONENTES	50
13. CONCLUSIONES	53
14. TRABAJO A FUTURO	54
15. REFERENCIAS	55
16. ANEXOS.....	56
16.1 HOJA DE ESPECIFICACIONES SENSOR MPU6050	56
16.2 HOJA DE ESPECIFICACIONES SENSOR HUELLA DIGITAL	57
16.3 HOJA DE ESPECIFICACIONES POTENCIÓMETRO	58
16.4 HOJA DE ESPECIFICACIONES PUENTE HLM293D	58

4. Índice de ilustraciones

<i>Figura 1 Sistema Rodanliu OSBS001</i>	<i>11</i>
<i>Figura 2 Encendido de motor por huella dactilar de Hyundai</i>	<i>12</i>
<i>Figura 3 Huella Digital</i>	<i>13</i>
<i>Figura 4 Funcionamiento interno de un sensor MEMS</i>	<i>14</i>
<i>Figura 5 Esquema de bloques de un procesador digital secuencial síncrono (13)</i>	<i>16</i>
<i>Figura 6 Diagrama arquitectura Von Neumann (13)</i>	<i>17</i>
<i>Figura 7 Diagrama arquitectura Harvard (13)</i>	<i>17</i>
<i>Figura 8 Estructura pantalla LCD</i>	<i>18</i>
<i>Figura 9 Circuito puente H</i>	<i>20</i>
<i>Figura 10 Diagrama de inversión de giro</i>	<i>20</i>
<i>Figura 11 Sensor Huella digital A071405</i>	<i>22</i>
<i>Figura 12 Giroscopio ITG32000.....</i>	<i>23</i>
<i>Figura 13 Acelerometro MPU6050</i>	<i>23</i>
<i>Figura 14 Arduino y su entorno</i>	<i>24</i>
<i>Figura 15 IDE de Arduino</i>	<i>25</i>
<i>Figura 16 Display LCD</i>	<i>27</i>
<i>Figura 17 Motores DC de espejos</i>	<i>29</i>
<i>Figura 18 Puente H Lm293D</i>	<i>30</i>
<i>Figura 19 Posición sensor MPU6050.....</i>	<i>35</i>
<i>Figura 20 PinOut sensor MPU6050</i>	<i>38</i>
<i>Figura 21 Conexión sensor MPU6050 a Arduino</i>	<i>38</i>
<i>Figura 22 Lectura de datos sensor MPU6050</i>	<i>40</i>
<i>Figura 23 PinOut Sensor huella digital.....</i>	<i>41</i>
<i>Figura 24 Conexión sensor huella digital a Arduino</i>	<i>41</i>
<i>Figura 25 Prueba de comunicación de sensor huella digital con Arduino</i>	<i>42</i>
<i>Figura 26 Prueba toma de imagen de huella digital</i>	<i>44</i>
<i>Figura 27 PinOut display LCD</i>	<i>45</i>
<i>Figura 28 Conexión de display LCD a Arduino.....</i>	<i>46</i>
<i>Figura 29 Visualización de instrucciones en display LCD</i>	<i>47</i>
<i>Figura 30 PinOut de alimentación de Arduino</i>	<i>48</i>
<i>Figura 31 Polaridad de conector jack</i>	<i>48</i>
<i>Figura 32 Alimentación de Arduino y Motores con misma fuente</i>	<i>49</i>
<i>Figura 33 Banco de prueba</i>	<i>51</i>
<i>Figura 34 Componentes del banco de prueba</i>	<i>52</i>

5. Índice tablas

<i>Tabla 1 Abreviatura</i>	7
<i>Tabla 2 Comparativa microcontroladores</i>	26
<i>Tabla 3 Comparación puente H</i>	30
<i>Tabla 4 Tabla de verdad con 1 motor</i>	33
<i>Tabla 5 Tabla de verdad con 2 motores</i>	34
<i>Tabla 6 Especificaciones sensor MPU6050</i>	36
<i>Tabla 7 Registro acelerómetro (11)</i>	36
<i>Tabla 8 Valores AFS_SEL (11)</i>	37
<i>Tabla 9 Registro giroscopio (11)</i>	37
<i>Tabla 10 Valores FS_SEL (11)</i>	37

6. Índice diagrama de flujo

<i>Diagrama de flujo 1 Funcionamiento general</i> _____	31
<i>Diagrama de flujo 2 Control de espejos</i> _____	32

7. Índice línea de código

<i>Línea de código 1 Movimiento del motor</i>	33
<i>Línea de código 2 Movimiento espejos manual</i>	34
<i>Línea de código 3 Configuración registros MPU6050</i>	37
<i>Línea de código 4 Selección de sensor MPU6050</i>	39
<i>Línea de código 5 Filtro sensor MPU6050</i>	39
<i>Línea de código 6 Almacenamiento huella digital</i>	43
<i>Línea de código 7 Display LCD</i>	46

8. Índice abreviaturas

Abreviatura	Significado
I²C	Circuito integrado interno (<i>Inter-Integrated Circuit</i>)
LED	Diodo emisor de luz (<i>Light Emitting Diode</i>)
LCD	Dsisplay líquido de cristal (<i>Liquid Crystal Display</i>)
UART	Emisor-Receptor asincrono universal (<i>Universal Asynchronous Receiver-Transmitter</i>)
SPI	Interfaz periférica de serie (<i>Serial Peripheral Interface</i>)
TTL	Transistor – Lógica de transistor (<i>Transistor-Transistor Logic</i>)
SCL	Reloj del sistema (<i>System Clock</i>)
SDA	Dato del sistema (<i>System Data</i>)
DC	Corrinete directa (<i>Direct Current</i>)
RX	Receptor (<i>Reciever</i>)
TX	Transmisor (<i>Transmission</i>)
ADC	Convertidor digital a analógico (<i>Analog to Digital Converter</i>)
LSB	Bit menos significativo (<i>Least Significant Bit</i>)

Tabla 1 Abreviatura

9. Introducción

9.1 Planteamiento del problema

En 2016 en México había más de 42 millones de automóviles circulando y ahora en el 2017 hay un poco más de 41 millones (INEGI, 2016), lo cual significa que las personas comparten más el automóvil para evitarse el tráfico y a la vez poder contribuir con el medio ambiente. Por lo tanto, como un mismo automóvil es ocupado por varias personas, estas andan acomodando los espejos cada vez que se suben al auto, ya que el usuario anterior lo dejó en diferente posición.

El tener en su correcta posición los espejos es de vital importancia para la comodidad del conductor y para la seguridad vial, como la mayoría que vive en la Ciudad de México tiene prisa, no se fija si sus espejos están en la correcta posición y cuando va manejando es el momento en que se da cuenta que están mal posicionados, por intentar acomodarlos en el camino la persona se expone a un accidente.

También hoy en día se necesita de más seguridad en un carro, ya que hay muchos robos en México, en el 2017 aumentaron los robos de autos en un 27.2% según el director general de la Asociación Mexicana de Instituciones de Seguros (AMIS), Recaredo Arias Jiménez [9].

9.2 Objetivo general

Realizar un banco de prueba en el cual se pueda observar el movimiento de los espejos y a la vez simular el encendido del motor mediante el sensor de huella digital.

9.3 Objetivos particulares

- Diseñar y construir un banco de prueba para los espejos.
- Mediante un LED y en el LCD dar aviso de que el motor fue encendido correctamente.
- Construir la interfaz del sistema en general donde se mostrarán las instrucciones.

9.4 Justificación

En la actualidad los sistemas de seguridad mediante huella digital se ocupan en accesos a habitaciones confidenciales, en cajas fuertes, en checadores, etc. Por lo tanto, implementar este sistema de seguridad en un automóvil impactaría en México por la inseguridad que hay en el país y la cantidad de robos de automóviles, con esto se podría llegar a reducir un porcentaje de robos en vehículos que cuenten con este sistema.

Otra problemática que hay al usar el automóvil, es el de tener un accidente por no estar atento a los alrededores y al dar una vuelta o cambiar de carril lo hacen sin ver adecuadamente a través de los espejos laterales.

A la hora de manejar es muy importante estar atento siempre a los espejos ya que te permiten ver quién está detrás de nosotros, incluso para hacer una maniobra de emergencia. Los espejos laterales son nuestros ojos para ver los puntos ciegos y nos ayudan a corroborar si vamos adecuadamente dentro del carril, por lo tanto, se deben de mantener en buena posición, la cual se tiene que ver la manija de la parte trasera, con esto se sabrá que está en una buena posición (El Universal, 2017).

Para resolver estas problemáticas, se requiere de tecnologías para ayudar a los conductores a tener un viaje más cómodo y seguro con un simple toque.

9.5 Antecedentes

9.5.1 Huella digital Seguridad automotriz Sistema (Rodanliu OSBS001)

Es un sistema de arranque de coche con huella dactilar, el cual cuenta con un sensor fabricado por FPC company. Este sistema es adaptable a cualquier automóvil.

Su función principal es identificar huellas dactilares para encriptar el sistema de antirrobo del vehículo, eliminando la tecnología de control de alarma original del coche.

Para iniciar el sistema se debe insertar las llaves del coche en el interruptor y ponerlo en "ON" e introducir la huella digital correcta, el coche desarmará la alarma y arrancará. A continuación, se muestra el diagrama de flujo para el inicio del sistema.

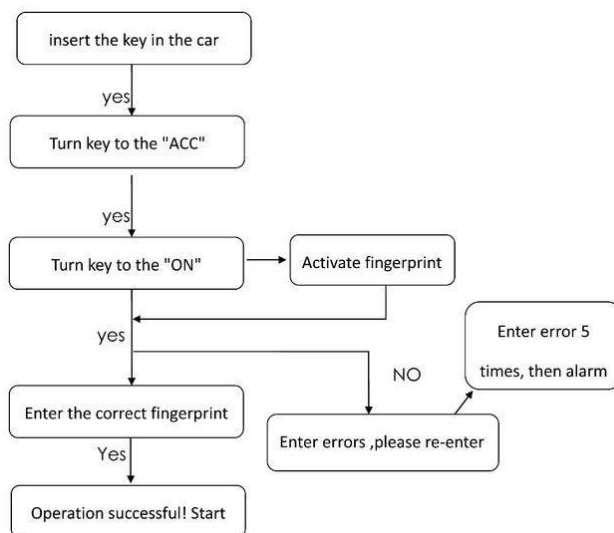


Diagrama de flujo 1Funcionamiento sistema Rodanliu OSBS001

Su configuración es mediante botones donde seleccionas almacenar o eliminar huellas dactilares rápidamente. Para eso es necesario ingresar la contraseña que se eligió anteriormente.



Figura 1 Sistema Rodanliu OSBS001

De esta forma es como llega el sistema y así es como se ve una vez adaptado al automóvil.

9.5.2 Sistema de huella dactilar para encender motor de Hyundai

La tecnología es apenas un prototipo que propone que el conductor coloque un dedo en el sensor ubicado en la manija de la puerta para abrir el auto.

También permitirá arrancar el motor con un sensor equipado en el interior del auto. Lo anterior, busca eliminar el uso de llaves físicas.



Figura 2 Encendido de motor por huella dactilar de Hyundai

Albert Biermann, presidente y director de la División de Investigación de Hyundai a nivel global, dijo este lunes que la surcoreana busca expandir la aplicación de la tecnología para permitir el ajuste de la temperatura, la posición del volante y otras características con la intención de personalizar la experiencia de manejo.

10. Marco teórico

En el marco teórico se hablará de los componentes que se usaran en el presente proyecto.

10.1 Huella digital

La biometría es el reconocimiento de un individuo usando ciertas características físicas y/o de comportamiento. Las huellas dactilares se forman alrededor de los 7 meses del desarrollo del feto y la configuración de los bordes no cambia durante la vida del individuo a menos que sea por una cortada o raspadura en las yemas, pero al cabo de 15 días esta se puede regenerar con la misma forma.



Figura 3 Huella Digital

Por lo tanto, la probabilidad de que 2 huella dactilares coincidan es de 1.9×10^{-15} . Hay 2 métodos para para procesar las huellas dactilares: características locales y patrones globales. El primero analiza las bifurcaciones y terminaciones de los bordes, el segundo representa una aproximación macroscópica ya que considera el flujo de los bordes como los arcos, lazos y espirales.

10.2 Giroscopio

Son dispositivos que miden la velocidad angular con la cual también puedes saber la posición de algún objeto. Cuando un objeto gira alrededor de un eje se obtiene una velocidad angular, la cual puede ser medida en revoluciones por segundo (RPS) o en grados por segundo ($^{\circ}/s$). Por lo general los giroscopios tienen una interfaz SPI o I2C, esto permite que sea más fácil la comunicación con un microcontrolador.

La sensibilidad se mide en mV por grado segundo. Se determina la cantidad de cambios de voltaje, por ejemplo, si un giroscopio que tiene una sensibilidad de $30 \text{ mV}/^{\circ} \text{ s}$ y se ve un cambio de 300 mV en la salida quiere decir que este giroscopio rota a $10 \text{ }^{\circ}/s$.

Este tipo de giroscopios cuentan con un sensor MEMS (sistemas microelectromecánicos) el cual es muy pequeño. Cuando se hace girar el giroscopio, una pequeña masa de resonancia se desplaza con los cambios de velocidad angular. Este movimiento se convierte en señales eléctricas de muy bajas corrientes que se pueden amplificar para ser leídas por un microcontrolador.

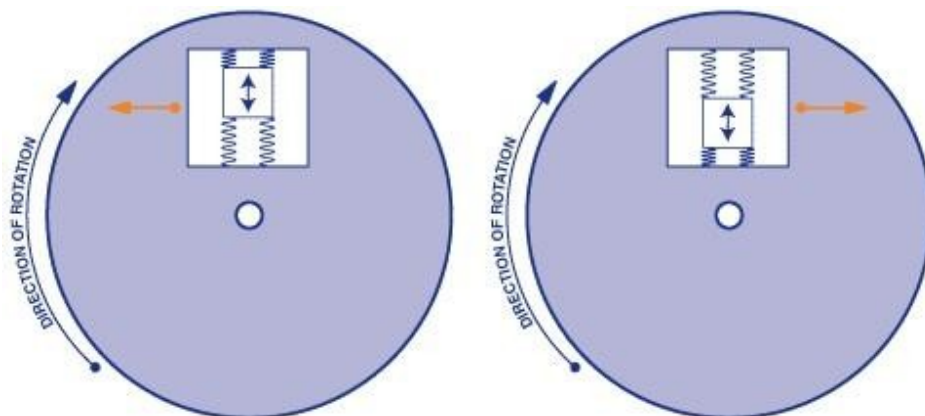


Figura 4 Funcionamiento interno de un sensor MEMS

Los giroscopios MEMS son generalmente dispositivos de menos potencia. Las corrientes de operación están en el orden de los miliamperes y a veces los micro Ampere. El voltaje de fuente de los giroscopios es generalmente de 5V o menos. Los giroscopios digitales pueden tener voltajes lógicos seleccionables y operan en el voltaje de fuente.

10.3 Microcontrolador

Un microcontrolador es un circuito integrado digital monolítico que contiene todos los elementos de un procesador digital secuencial síncrono programable de arquitectura Harvard o Princeton. Se le puede denominar también microcomputador integrado y esta especialmente orientado a tareas de control y comunicaciones.

Por su pequeño tamaño, los microcontroladores permiten empotrar un procesador programable en muchos productos industriales. Su costo reducido y su consumo de energía y velocidad adaptables resultan apropiados para numerosas aplicaciones.

Los procesadores digitales secuenciales tienen asociados 3 conceptos principales:

- Arquitectura interna
- Juego de instrucciones
- Interfaces de entradas y salidas

10.3.1 Arquitectura interna

Es el conjunto de atributos que tienen un impacto directo en la ejecución del proceso que llevan a cabo. A continuación, se muestra el diagrama de bloques de un procesador secuencial síncrono, formado por una unidad de control y una unidad operativa, compuesta por una unidad de memoria y una unidad aritmética y lógica

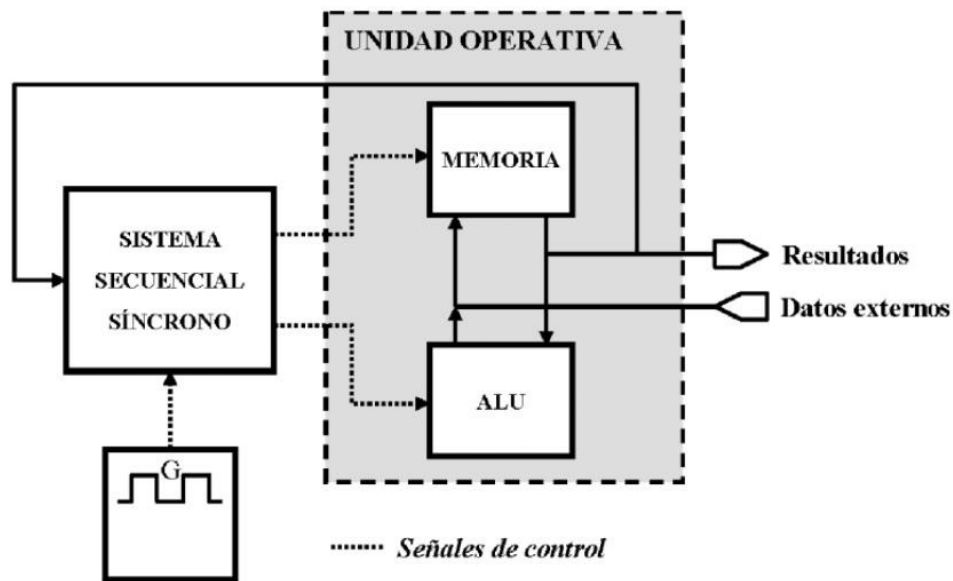


Figura 5 Esquema de bloques de un procesador digital secuencial síncrono (13)

Los datos externos se memorizan en los registros adecuados de la unidad de memoria para presentarlos en secuencia en las entradas de la unidad aritmética y lógica, que los procesa y los devuelve la unidad de memoria los resultados parciales y finales obtenidos.

La arquitectura de un procesador digital está directamente relacionada con la forma de implementar la unidad de control. Existen 2 formas de llevar a cabo dicha implementación, las cuales son los procesadores digitales con arquitectura Harvard y los de arquitectura Von Neumann.

10.3.2 Arquitectura Von Neumann

La arquitectura tradicional de computadoras y microcontroladores se basa en el esquema propuesto por John Von Neumann, en el cual la unidad central de proceso, o CPU, esta conectada a una memoria única que contiene las instrucciones del programa y los datos. El tamaño de la unidad de datos o instrucciones esta fijado por el ancho del bus de la memoria.

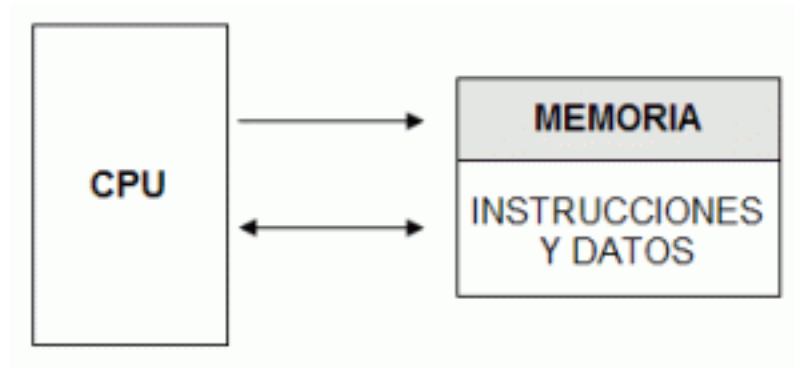


Figura 6 Diagrama arquitectura Von Neumann (13)

10.3.3 Arquitectura Harvard

Este tipo de estructura se caracteriza por tener como unidad de control un sistema secuencial síncrono que posee un contador de entrada y salida en paralelo. También cuenta con una memoria para datos y una para instrucciones, de esta manera se utilizan 2 buses diferentes. Con esto se permite trabajar con las 2 memorias al mismo tiempo, siendo así que sea mas rápida la ejecución de programas.

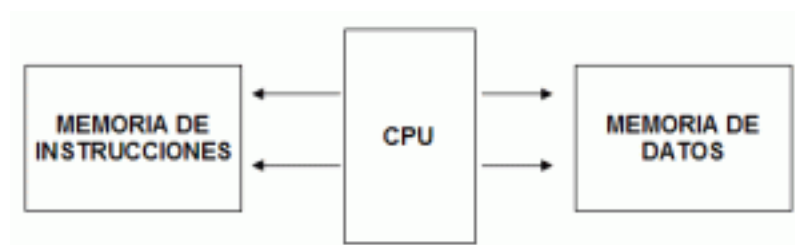


Figura 7 Diagrama arquitectura Harvard (13)

10.4 Display LCD

En nuestra vida cotidiana estamos rodeados de advertencias visuales, esto debido a que nos facilitan a entender el funcionamiento de algún sistema o incluso para saber hacia dónde debemos de ir. Por lo cual es muy importante contar con un algún tipo de pantalla que nos indique que se debe de hacer en ese sistema nuevo que estamos manipulando.

Y uno de los tipos de pantalla mas usado son los de LCD (Liquid Crystal Display), el cual es una pantalla plana formado por pixeles en color o monocromáticos colocados frente una fuente de luz. Este tipo de pantalla consume cantidades muy pequeñas de energía.

Cada píxel de un LCD típicamente consiste en una capa de moléculas alineadas entre dos electrodos transparentes, y dos filtros de polarización, los ejes de transmisión de cada uno que están perpendiculares entre sí. Sin cristal líquido entre el filtro polarizante, la luz que pasa por el primer filtro sería bloqueada por el segundo polarizador.

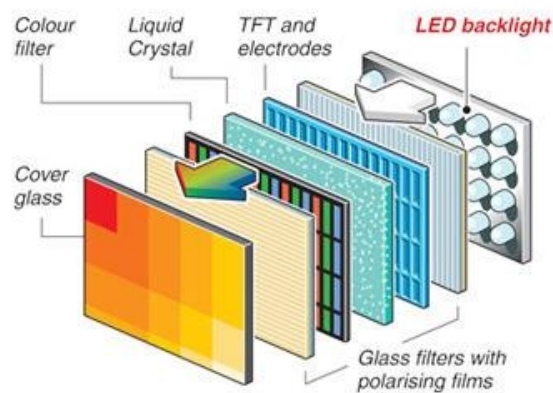


Figura 8 Estructura pantalla LCD

Alguno de las características que se deben tomar en cuenta son:

1. **Resolución:** Las dimensiones son horizontales y verticales que se miden en pixeles.
2. **Tiempo de respuesta:** Es el tiempo que tarda de cambiar un píxel de un color a otro.
3. **Brillo:** Cantidad de luz emitida desde la pantalla.
4. **Contraste:** La relación de intensidad entre la mas brillante y la mas obscura.
5. **Tamaño:** El tamaño se mide de forma diagonal y se expresa en pulgadas.

10.5 Puente H

El puente H es un circuito electrónico que permite a un motor eléctrico DC girar en ambos sentidos, avanzar y retroceder.

Los puentes H ya vienen hechos en algunos circuitos integrados, pero también se pueden construir a partir de componentes eléctricos y/o electrónicos.

Un puente H se construye con 4 interruptores (mécánicos o mediante transistores). Cuando los interruptores S1 y S4 están cerrados (S2 y S3 abiertos) se aplica una tensión haciendo girar el motor en un sentido. Abriendo los interruptores S1 y S4 (cerrando S2 y S3), el voltaje se invierte, permitiendo el giro en sentido inverso del motor.

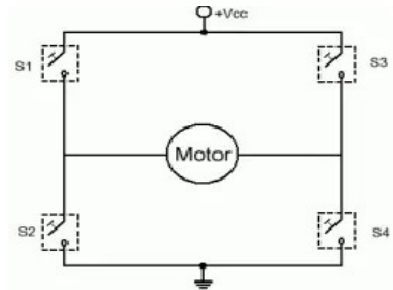


Figura 9 Circuito puente H

Cuando los interruptores S1-S4 (según la figura) están cerrados (S2-S3 estarán abiertos), el motor será atravesado por la tensión en una dirección, a la que girará el motor. Ahora bien, si abrimos S1-S4

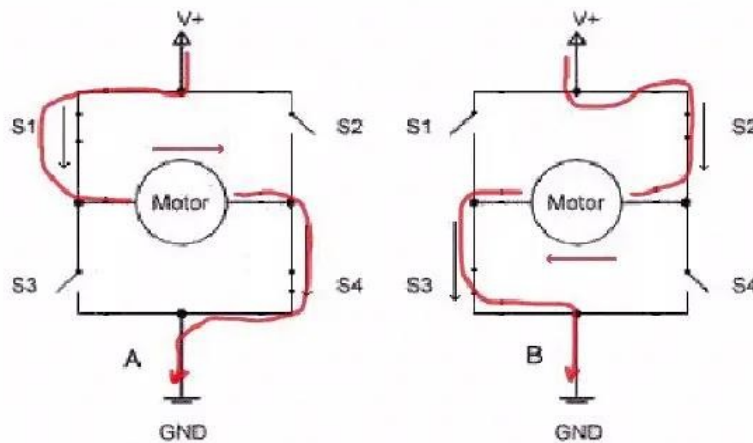


Figura 10 Diagrama de inversión de giro

Si siguiendo la nomenclatura anterior, los interruptores S1-S3, nunca deberán ser cerrados al mismo tiempo, esto causaría un cortocircuito en la fuente de tensión de entrada. Lo mismo, ocurre con los interruptores S2-S4.

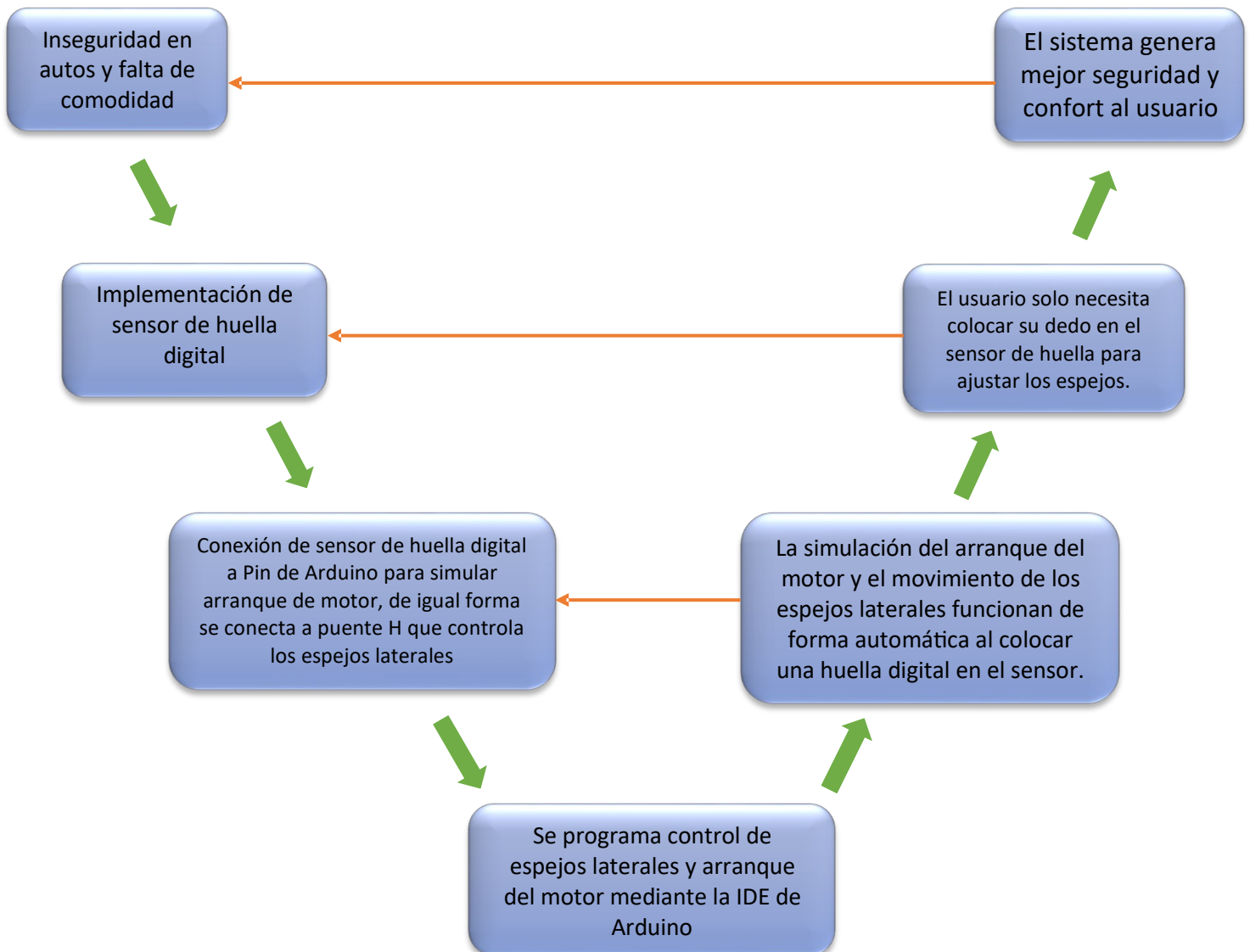
Si los interruptores S1, S2, S3 y S4 están abiertos, el motor estará libre, en cambio si el motor está girando y queremos frenarlo, la tensión inducida que se genera con la inercia se debe derivar hacia un sistema llamado de frenado dinámico.

En general, los cuatro elementos de conmutación se pueden activar y desactivar de forma independiente.

11. Desarrollo

Durante el desarrollo del proyecto se debieron de tener a consideración características específicas de cada componente para tener la elección correcta de los sensores, display y controladores que sean los mas aptos para el sistema.

11.1 Diagrama Metodología V



11.2 Sensor Huella digital

En la industria existen varios lectores de huellas digitales para diversas aplicaciones, desde una industrial hasta un sensor para estudiantes, como lo es el sensor 071405.



*Figura 11 Sensor Huella digital
A071405*

Las características de dicho sensor son las siguientes:

- Modelo: 071405
- Voltaje de alimentación: 3.6V - 6V
- Corriente de operación: 100mA - 150mA
- Interfaz: UART TTL
- Modo de paridad de huella: 1:1 1:N
- Baud Rate: 9600*N
- N = 1 a 12 (Por defecto es 6)
- Tiempo de adquisición menor a 1 segundo
- Niveles de seguridad
- Dimensión de la ventana: 14x18mm
- Entorno de trabajo: -10°C a 40°C (Humedad Relativa 40% a 85%)
- Dimensiones: 5.5 x 2.1 x 2.0 cm
- Peso: 22g

Este sensor es perfecto para proyectos escolares ya que es compatible con Arduino, es accesible, fácil de conseguir y económico a comparación de otros sensores biométricos.

11.3 Sensor MPU6050

Existen diversos tipos de giroscopios como el ITG3200, el cual es un integrado que cuenta con un giroscopio MEMS como se muestra en la figura 3.

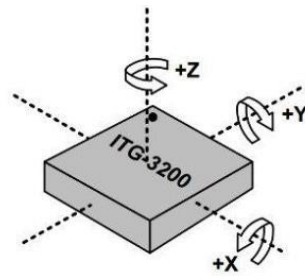


Figura 12 Giroscopio ITG32000

Sus características son:

- Ejes 3 (X,Y,Z)
- Rango $\pm 2000^\circ/\text{s}$
- Interfaz Digital I²C-16 bits
- Alimentación 2.1V a 3.6V 6.5 mA
- Filtro pasa baja programable.
- Sensor de temperatura.

También está el módulo MPU-6050 el cual cuenta con 3 ejes y un acelerómetro

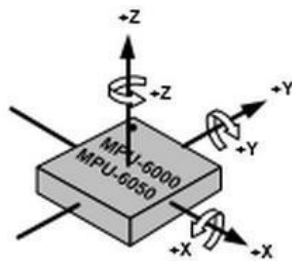


Figura 13 Acelerometro MPU6050

Sus características son:

- Alimentación: 3-5V
- Comunicación: Digital I²C.
- Convertidor analógico a digital de 16 bit en todos los ejes.
- El giroscopio dispone de cuatro escalas programables por el usuario: ± 250 , ± 500 , ± 1000 y ± 2000 ° / seg (dps).
- La escala del acelerómetro es programables por el usuario con valores de $\pm 2g$, $\pm 4g$, $\pm 8g$ y $\pm 16g$.
- Sensor de temperatura integrado.
- Filtro paso bajo programable.

11.4 Arduino

Es una plataforma de hardware libre, consta de un microcontrolador ATmega2560 y un entorno de desarrollo el cual fue diseñado para realizar proyectos de electrónica multidisciplinarios.



Figura 14 Arduino y su entorno

Arduino proporciona un software consistente en un entorno de desarrollo IDE el cual implementa el lenguaje de programación de Arduino. Una característica importante de este lenguaje de programación es su sencillez de uso. Como podemos observar en la figura 5, se necesitan de sensores y actuadores para que el Arduino realice trabajos mediante la programación que nosotros generamos.

Esta tarjeta tiene varias características, las principales son:

- Microcontrolador ATmega2560.
- Voltaje de entrada 7-12V.
- 54 entradas digitales de I/O (15 salidas PWM).
- 16 entradas análogas.
- 32k de memoria Flash.
- Reloj de 16MHz de velocidad.

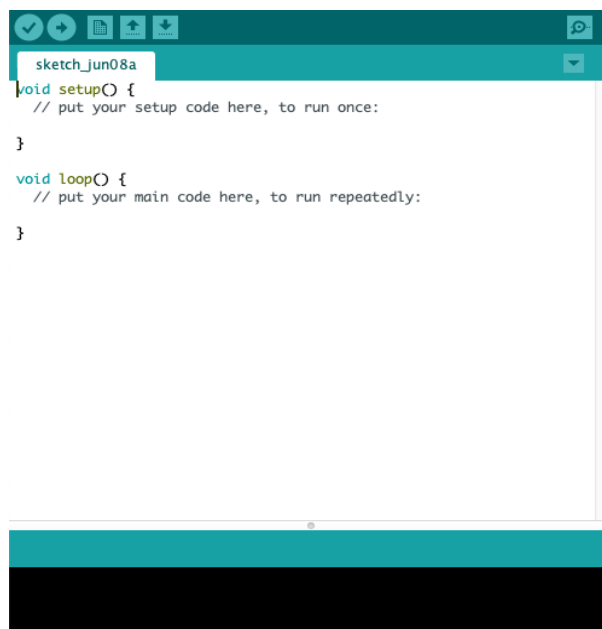


Figura 15 IDE de Arduino

<i>Característica</i>	Arduino MEGA	Raspberry Pi 3	ST núcleo STM32
<i>Velocidad</i>	16MHz	1.2 GHz	84MHz
<i>SRAM</i>	8KB	1GB	96KB
<i>Memoria Flash</i>	256KB	N/A (se necesita comprar aparte)	512KB
<i>Voltaje entrada</i>	5V - 12V	5.1V DC	3.3V, 5V, 7V a 12V
<i>Entradas digitales</i>	54 I/O	40 I/O	50 I/O
<i>Entradas analógicas</i>	16	N/A	N/A
<i>Lenguaje programación</i>	C++, Arduino	C++, Phyton	C++,
<i>Conocimiento</i>	85%	30%	50%
<i>Costo</i>	\$270 (se cuenta con el)	\$890	\$350 (se cuenta con el)

Tabla 2 Comparativa microcontroladores

La selección de este microcontrolador fue basada en la siguiente tabla.

Las características de cada tarjeta que se tomaron en cuenta para la comparativa fueron en base a las funciones que necesitaremos para el sistema de encendido y la automatización de los espejos.

Por ejemplo, nos interesa la alimentación que necesita la tarjeta para funcionar correctamente, entonces al comparar las 3 tarjetas nos damos cuenta de que tanto el Arduino como la STM32 pueden ser alimentados desde 5V hasta 12V, mientras que la Raspberry solo se alimenta con 5V. En el automóvil el voltaje que sale de la batería es de 12V, por lo que la Raspberry por sí sola no puede ser

alimentada, se necesitaría de un regulador mientras que las otras 2 las podemos conectar directamente.

De la SRAM realmente no se ocupa mucha, ya que el programa no es de muchas instrucciones, por lo que con los 8kB que tiene el Arduino son más que suficientes, aunque las demás tengan más memoria no es algo indispensable que necesitemos para el funcionamiento correcto del programa.

El lenguaje de programación para los 3 puede ser el mismo, C++, pero tenemos la ventaja con la IDE de Arduino que es más amigable y se tiene mayor conocimiento sobre ese lenguaje.

En conclusión, el Arduino es el que más se adapta al sistema y a los recursos con los que se cuenta, ya que al elegir la Raspberry nos implica más gasto tanto por necesidad de comprar una tarjeta y por la inversión de tiempo para aprender a usarla de mejor manera, pero no es necesaria para que el sistema funcione de mejor manera.

11.5 Display LCD

En este proyecto se decidió usar un Display LCD para dar las indicaciones que debe de seguir el usuario para poder encender el automóvil y ajustar los espejos laterales. El display que se uso fue el de 16x2, ya que no son muchas las instrucciones que se necesitan dar para usar de forma correcta el sistema y



Figura 16 Display LCD

debido al pequeño tamaño se puede adaptar prácticamente en cualquier parte del automóvil.

Las características principales de este tipo de display son las siguientes:

- 1) El **tamaño** de 2x16 se refiere a que puedes colocar 16 caracteres al mismo tiempo en 2 renglones de manera horizontal.
- 2) La **resolución** es de 128x64 con un fondo verde y caracteres negros.
- 3) Cuenta con un **brillo** que puede ser variable para ajustarlo al entorno que se encuentra.
- 4) Cuenta con una **iluminación** LED las cuales están en los bordes del display, esta iluminación puede ser variable dependiendo el valor de la resistencia que le coloquemos.
- 5) El **contraste** del display se lo podemos ajustar con un potenciómetro externo.

Una vez ajustando todos estos valores a las necesidades que tengamos, el usuario podrá ver de manera clara las instrucciones.

Este tipo de display tiene un controlador Hitachi HD44780, el cual nos permite que conectemos directamente el LCD directo al Arduino, pero se necesitan varios pines del Arduino (12pines) para hacer la comunicación. Es por eso por lo que existen otras 2 formas de comunicación, pero estas necesitan de un módulo externo para poder lograr el intercambio de datos.

El primero es mediante UART, con este se ocupan los pines Rx y Tx y nos reduce el uso de solo 4 pines.

La segunda forma es mediante I²C, Este ocupa los pines de SCL y SDA y de igual forma, nos reduce el uso de 4 pines del Arduino.

Prácticamente, tenemos 3 formas de comunicar el Arduino con el LCD solo es cuestión de elegir el que mejor se adapte a nuestro sistema.

11.6 Puente H Lm293D

Los espejos cuentan con 2 motores DC y un juego de engranes que permiten que el espejo se mueva en 4 direcciones, como se puede observar en la siguiente imagen:



Figura 17 Motores DC de espejos

Para hacer el control, primero se tuvo que usar un puente H para permitir que el motor vaya en el sentido que nosotros necesitamos.

Y para poder elegir el puente H que más se nos adapte a nuestro sistema, debemos poner atención en las siguientes especificaciones:

- Voltaje de entrada -> Para saber con que voltaje se cuenta para poder alimentarlo.
- Consumo mínimo para operación optima -> Conocer cuanto amperaje nos consumirá de nuestra alimentación primaria.
- Salida de voltaje y corriente de motores -> Tener en cuenta La cantidad de voltaje y corriente que consumen los motores.
- Cantidad de motores que se pueden controlar-> Considerar que se tienen 2 motores por espejo, es decir, se necesitan controlar 4 motores.

- Necesidad de componentes externos al puente H -> Revisar si se necesita comprar componentes a parte del puente H

A continuación, se vera la comparativa de los puentes H:

<i>Característica</i>	Lm293D	Transistores	Módulo L298N
<i>Voltaje entrada</i>	5V	5V	5V
<i>Salida de voltaje</i>	5V -36V	5V-36V	5V- 36V
<i>Número motores que controla</i>	2	1	2
<i>Componentes externos</i>	SI	SI	NO
<i>Costo</i>	\$40	\$35	\$70

Tabla 3 Comparación puente H

En esta ocasión se usó el integrado lm293D debido a que tiene la posibilidad de controlar 2 motores y es de tamaño reducido, la conexión de este integrado es de la siguiente forma:

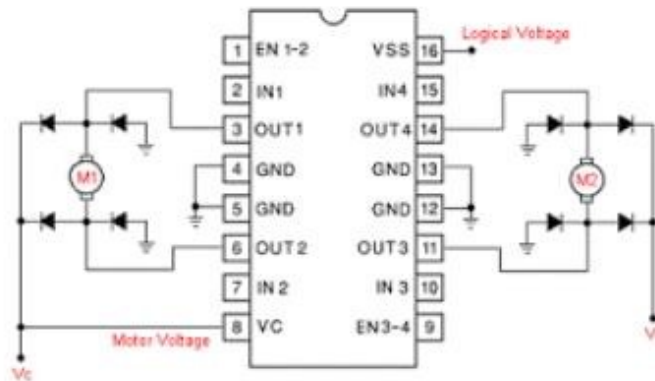


Figura 18 Puente H Lm293D

12. Resultados

12.1 Funcionamiento general

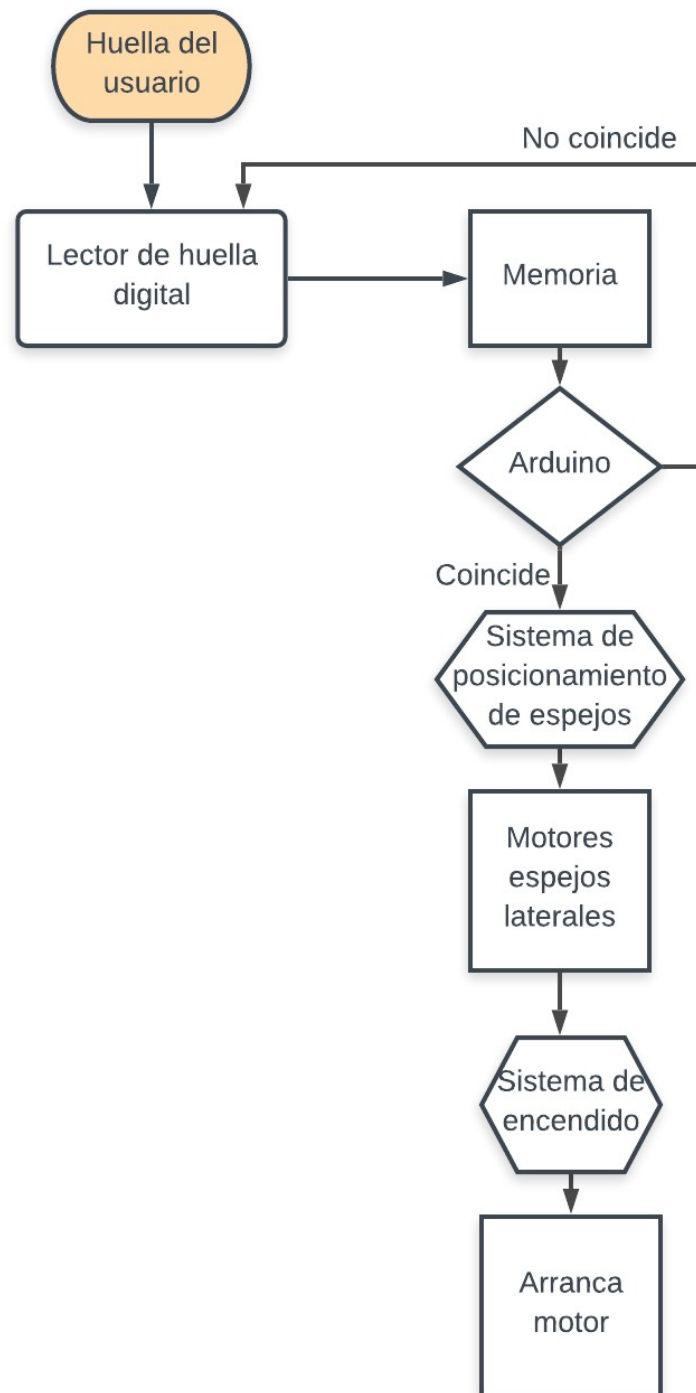


Diagrama de flujo 2 Funcionamiento general

12.2 Control espejos

La lógica del funcionamiento del espejo para estar en la posición deseada por el usuario es la siguiente:

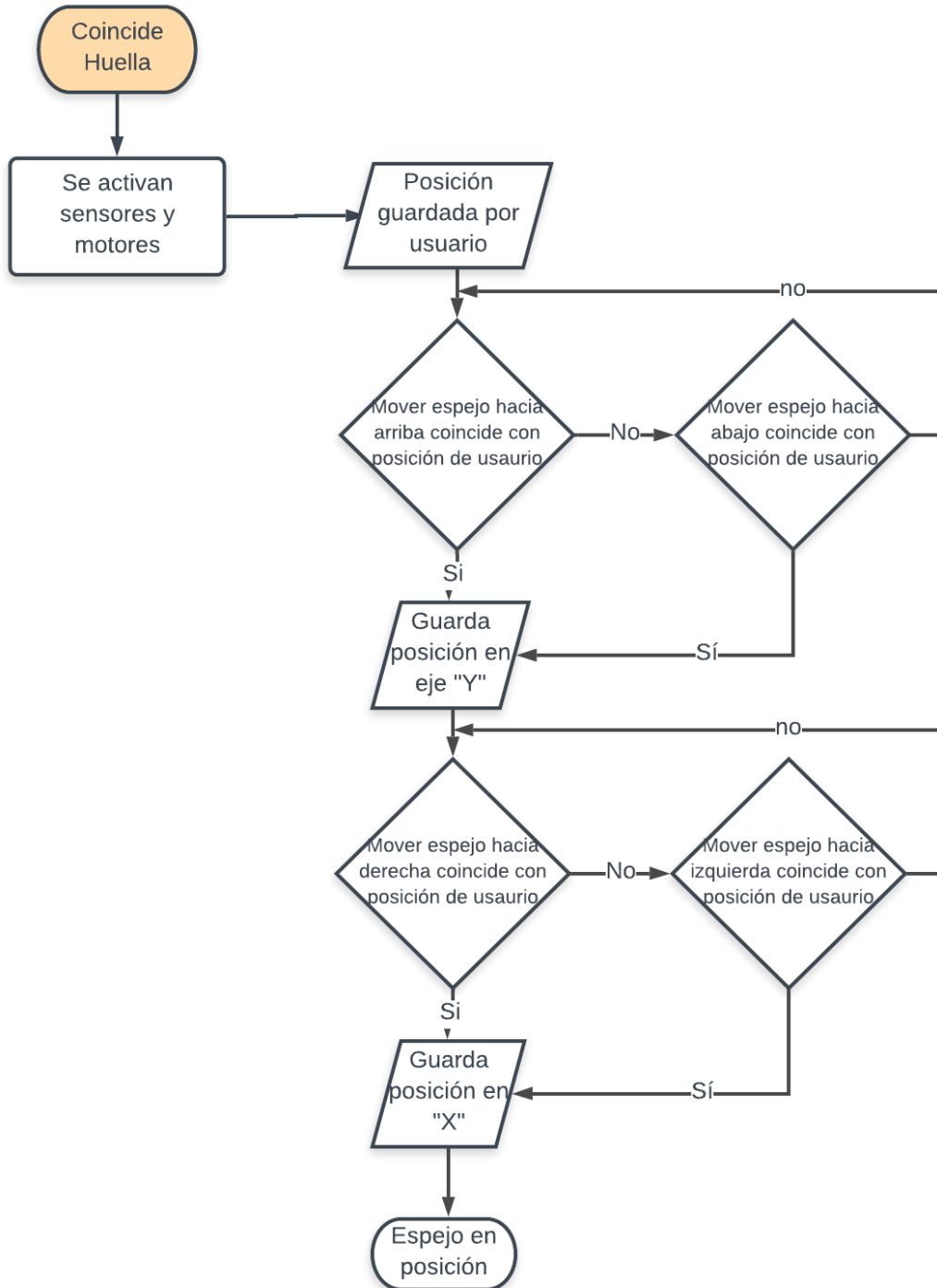


Diagrama de flujo 3 Control de espejos

12.3 Puente H lm293D

Una vez seleccionado el puente H lm293D se empieza la conexión del sensor al Arduino, donde los pines 3, 6, 14 y 11 son las salidas que mandan el pulso al motor y los pines 2, 7, 15 y 10 son las entradas, como se puede observar en la figura 15, estos son los que van conectados al Arduino.

Con las siguientes líneas de programación se observa la forma en que se puede cambiar el sentido de los motores:

```
int motorpin1 = 8;  
int motorpin2 = 9;  
int motorpin3 = 10;  
int motorpin4 = 11;  
  
digitalWrite(motorpin1,LOW);  
digitalWrite(motorpin2,HIGH);  
digitalWrite(motorpin3,LOW);  
digitalWrite(motorpin4,HIGH);
```

Línea de código 1 Movimiento del motor

Lo primero que se hizo fue declarar los pines a los cuales irán las entradas del integrado, como se muestra en la primera imagen, después mediante un *digitalWrite* escribimos si se quiere que la salida sea alta o baja.

Motorpin1	Motorpin2	Salida
1	0	Giro Levógiro
0	1	Giro Dextrógiro
0	0	Freno a tierra
1	1	Freno a voltaje

Tabla 4 Tabla de verdad con 1 motor

Como se puede observar en la tabla, al alimentar una salida y dejar la otra sin voltaje nos genera un giro ya sea levógiro o dextrógiro, pero ahora al combinar los 2 motores que tiene el espejo, nos queda de la siguiente forma la tabla de verdad:

Motorpin1	Motorpin2	Motorpin3	Motorpin4	Salida
1	0	1	0	Giro a la derecha
1	0	0	1	Giro a la izquierda
0	1	1	0	Giro hacia arriba
0	1	0	1	Giro hacia abajo

Tabla 5 Tabla de verdad con 2 motores

Basándonos en la *tabla 4*, se pudo hacer la conexión y la programación necesaria para el movimiento de los espejos.

A parte del movimiento autónomo de los espejos también se hizo un control manual para que el usuario pueda seleccionar la posición que desea y posteriormente se guarde.

Para el control manual se usaron 2 potenciómetros uno para el movimiento vertical y el otro para el movimiento horizontal, a continuación, se muestra una sección del programa donde se puede explicar cómo se hace el control:

```

valor1 = analogRead(arribajo);
valor2 = analogRead(izder);
delay(50);
valor1_1 = analogRead(arribajo);
valor2_2 = analogRead(izder);
Serial.println(valor1_1);
if(valor1 < valor1_1){
digitalWrite(motorpin1, LOW);
digitalWrite(motorpin2, HIGH);
digitalWrite(motorpin3, LOW);
digitalWrite(motorpin4, HIGH);

```

Línea de código 2 Movimiento espejos manual

Una vez que hecho las pruebas correctas del movimiento de los motores de forma manual y de forma automática, se prosigue con la colocación del sensor que nos dará la posición del espejo, este es el MPU6050.



Figura 19 Posición sensor MPU6050

12.4 Sensor MPU6050

Para el uso de este sensor primero se debe de conocer bien que es lo que trae internamente y que valores son los que lee. A continuación, se hablarán de dichas especificaciones:

Empezaremos hablando sobre su alimentación, este sensor por si sólo necesita de 2.37V - 3.46V, lo cual es un problema ya que el Arduino nos da 5V de salida, pero el sensor que tenemos viene con un módulo en el cual tiene un regulador GY-521 permitiéndonos alimentarlo hasta con 5V.

El sensor contiene un giroscopio 3 ejes con el cual podemos medir velocidad angular y también cuenta con un acelerómetro igual de 3 ejes con los que

podemos medir la aceleración en X, Y, Z. A continuación, se mostrará una tabla mostrando los rangos de valores tanto del giroscopio como del acelerómetro, que se consultaron en el datasheet:

Rango de escala completa del giroscopio (°/s)	Sensibilidad del giroscopio LSB/(°/s)	Rango de escala completa acelerómetro (g)	Sensibilidad de acelerómetro (LSB/g)
± 250	131	± 2	16384
± 500	65.5	± 4	8192
± 1000	32.8	± 8	4096
± 2000	16.4	± 16	2048

Tabla 6 Especificaciones sensor MPU6050

Este sensor tiene rangos variables, lo cual nos permite elegir que tan preciso necesitamos que sea, es decir, si lo configuramos con una aceleración de ± 16g, tendremos una sensibilidad de 2048 por lo que a mayores rangos mayor lectura de aceleraciones grandes, pero es menos sensible a aceleraciones más pequeñas, estos valores están predeterminados a 16 bits, ya que este sensor cuenta con un ADC de 16 bits.

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1C	28	XA_ST	YA_ST	ZA_ST	AFS_SEL[1:0]				-

Tabla 7 Registro acelerómetro (11)

El registro “1C” es el que se encarga de accionar y configurar el acelerómetro, dándonos la opción de poder seleccionar que sensibilidad queremos utilizar, guiándonos de la tabla 3, podemos configurarlo dependiendo del uso que necesitemos, dicho valor se debe escribir en los bits 3 y 4 como se observa en la *tabla 5*, los valores que se necesitan poner para elegir la sensibilidad son los siguientes:

AFS_SEL	Full Scale Range
0	$\pm 2g$
1	$\pm 4g$
2	$\pm 8g$
3	$\pm 16g$

Tabla 8 Valores AFS_SEL (11)

Una vez elegido la sensibilidad debemos de poner en el AFS_SEL el valor que deseamos, en este caso se configuró con una escala de 8g, es decir que tenemos una sensibilidad de 4096 LSB/g.

De mismo modo se configura el giroscopio, solo que ese se escribe en el registro "1B".

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1B	27	XG_ST	YG_ST	ZG_ST	FS_SEL[1:0]		-	-	-

Tabla 9 Registro giroscopio (11)

En este caso nos basamos en los datos de la tabla 9, donde se escriben los valores en FS_SEL, siendo que se eligiera una escala de 100°/s, es decir, una sensibilidad de 32.8 LSB/(°/s)

FS_SEL	Full Scale Range
0	$\pm 250 \text{ } ^\circ/\text{s}$
1	$\pm 500 \text{ } ^\circ/\text{s}$
2	$\pm 1000 \text{ } ^\circ/\text{s}$
3	$\pm 2000 \text{ } ^\circ/\text{s}$

Tabla 10 Valores FS_SEL (11)

En la línea de código lo Podemos observar de la siguiente forma:

```
#define MPU6050_ACONFIG_AFS_SEL_BIT 4
#define MPU6050_ACONFIG_AFS_SEL_LENGTH 2
#define MPU6050_GCONFIG_FS_SEL_BIT 4
#define MPU6050_GCONFIG_FS_SEL_LENGTH 2
```

Línea de código 3 Configuración registros MPU605

Una vez ya configurada la sensibilidad que se requiere, procedemos a la programación de las condiciones, pero cabe destacar que para el sensor MPU6050 es necesaria la comunicación I²C para poder mandar los datos al Arduino, la ventaja que se tiene con este sensor es que te permite utilizar 2 direcciones para la comunicación ya sea la 0X68 ó la 0X69, se necesitaron las 2 debido a que se utilizó uno para el lado del conductor y otro para el lado del copiloto. Dicha selección de dirección se hace mediante código, donde lo veremos más adelante.

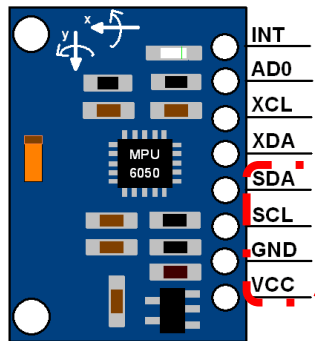


Figura 20 PinOut sensor MPU6050

La imagen anterior nos muestra los pines que tiene el sensor y cuáles son los que se usaran, tales como los pines SDA y SCL, estos son los que harán la comunicación I²C, por los que deben de ir conectados a los 20 (SDA) y 21 (SCL) del Arduino Mega, también se conectan los de alimentación (VCC, GND) y por último el pin AD0 se conecta a una salida digital para poder controlar la dirección deseada.

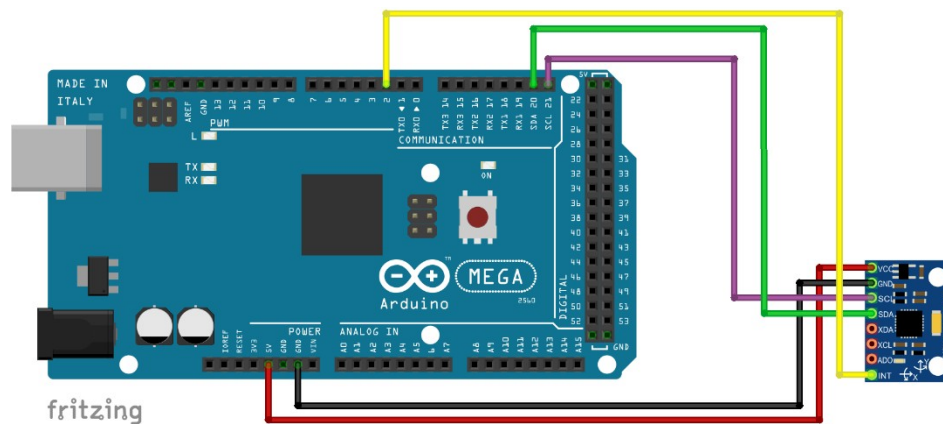


Figura 21 Conexión sensor MPU6050 a Arduino

En la siguiente parte del código se muestra como se hace la iteración entre sensores:

```
if (choose){
digitalWrite(MPUpinA0[12], LOW);
digitalWrite(MPUpinA0[13], HIGH);
//Sensores espejos
sensor.getAcceleration(&ax, &ay, &az);
sensor.getRotation(&gx, &gy, &gz);
}
```

Línea de código 4 Selección de sensor MPU6050

Cuando apretamos el botón para elegir el espejo, la dirección del sensor que ocuparemos se convierte en 0X69 y mientras que el otro sensor lo pone en bajo, a pesar de que se tienen 2 direcciones diferentes la lectura de datos debe de ser una por una, es decir, primero se debe de ajustar un sensor y una vez terminado se ajusta el otro.

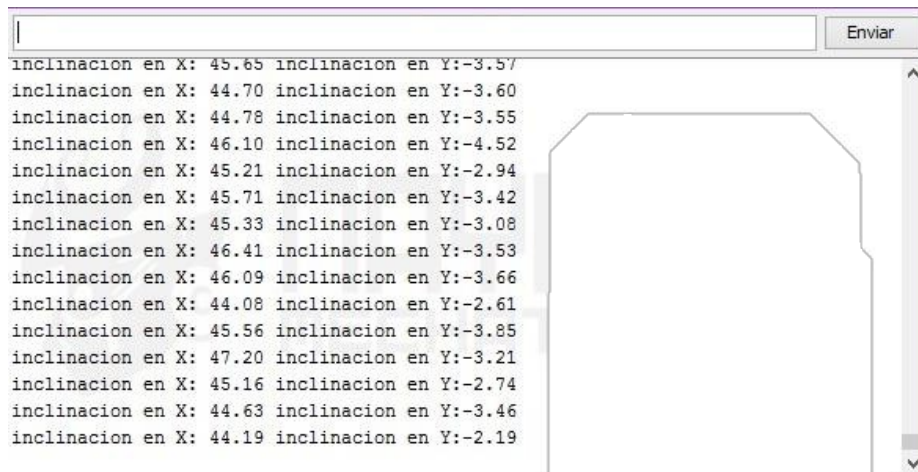
Debido a que este sensor es sensible, cualquier ruido, como las vibraciones afectan en la lectura de datos, por lo que se necesitara de un filtro para discriminar dichas interferencias. Se puede hacer uno físico o uno mediante programación. En esta ocasión se decidió hacer un filtro mediante programa debido a que es más confiable porque se basa en fórmulas matemáticas y a la vez tienes ahorro de material.

En esta sección del código se hacen los cálculos correspondientes para hacer el filtro Pasabajos.

```
float accel_ang_x2=atan(ay2/sqrt(pow(ax2,2) + pow(az2,2)))*(180.0/3.14);
float accel_ang_y2=atan(-ax2/sqrt(pow(ay2,2) + pow(az2,2)))*(180.0/3.14);
//Calcular angulo de rotación con giroscopio y filtro complemento
ang_x2 = 0.98*(ang_x_prev2+(gx2/131)*dt2) + 0.02*accel_ang_x2;
ang_y2 = 0.98*(ang_y_prev2+(gy2/131)*dt2) + 0.02*accel_ang_y2;
ang_x_prev2=ang_x2;
ang_y_prev2=ang_y2;
```

Línea de código 5 Filtro sensor MPU6050

Una vez programado el filtro y configurado la sensibilidad del acelerómetro y giroscopio, se empezaron con las pruebas con el puerto serial, como se observa a continuación, las lecturas que nos da el sensor son en el eje “x” y en el eje “y”.



```
inclinacion en X: 45.65 inclinacion en Y:-3.57
inclinacion en X: 44.70 inclinacion en Y:-3.60
inclinacion en X: 44.78 inclinacion en Y:-3.55
inclinacion en X: 46.10 inclinacion en Y:-4.52
inclinacion en X: 45.21 inclinacion en Y:-2.94
inclinacion en X: 45.71 inclinacion en Y:-3.42
inclinacion en X: 45.33 inclinacion en Y:-3.08
inclinacion en X: 46.41 inclinacion en Y:-3.53
inclinacion en X: 46.09 inclinacion en Y:-3.66
inclinacion en X: 44.08 inclinacion en Y:-2.61
inclinacion en X: 45.56 inclinacion en Y:-3.85
inclinacion en X: 47.20 inclinacion en Y:-3.21
inclinacion en X: 45.16 inclinacion en Y:-2.74
inclinacion en X: 44.63 inclinacion en Y:-3.46
inclinacion en X: 44.19 inclinacion en Y:-2.19
```

Figura 22 Lectura de datos sensor MPU6050

Como se puede observar, la lectura de datos tiene una diferencia de $\pm 2^\circ$ es por eso por lo que en el programa debemos de usar rangos para absorber esas variaciones, pero para saber que rangos son los que necesitaremos, primero debemos de colocar el sensor en la posición que va a quedar, como se observó en la *figura 9* y finalmente hacer la programación de los rangos.

12.5 Sensor de huella digital

Diagrama
Metología V

Para la implementación de dicho sensor en el proyecto, lo primero que se debe de hacer es la conexión adecuada para que tenga una buena comunicación el sensor con el Arduino.

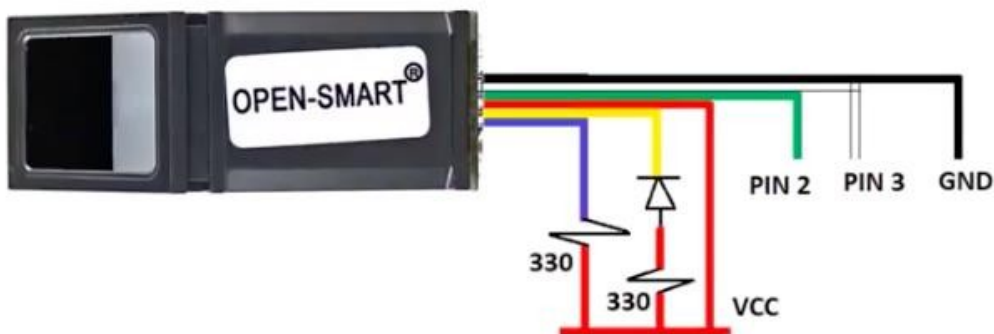


Figura 23 PinOut Sensor huella digital

Como vemos en la imagen anterior, los cables verde y blanco del sensor son el RX y la TX respectivamente, se usan estos pines ya que el sensor usa comunicación UART.

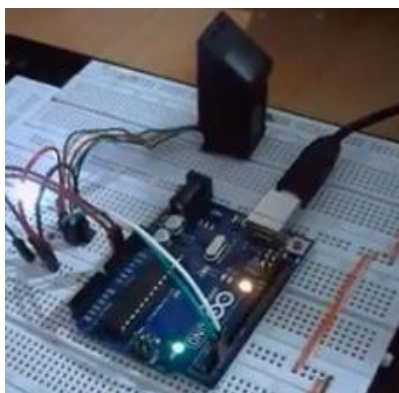


Figura 24 Conexión sensor huella digital a Arduino

Una vez conectado el sensor al Arduino se hicieron pruebas del funcionamiento, donde lo primero es establecer la comunicación correctamente, mediante el monitor serial podemos darnos cuenta si se conectó correctamente.

```
fingertest
Did not find fingerprint sensor :(
fingertest
Found fingerprint sensor!
Type in the ID # you want to save this finger as...
```

Figura 25 Prueba de comunicación de sensor huella digital con Arduino

Como observamos en la *figura 22*, nos indica que la conexión fue correcta y esta en espera de una huella. Una vez teniendo una buena comunicación lo que sigue es seleccionar en que memoria del sensor queremos que se guarde la huella, es decir, asignamos la huella a un número de usuario.

Finalmente, después de haber tenido la conexión correcta se hace el procedimiento para la lectura de la huella, la cual consiste en colocar el dedo en repetidas ocasiones en el sensor, pero para saber en que momento se debe de hacer se mostrara en el display LCD el momento en que le usuario debe de colocar su dedo en el cristal del sensor.

En la siguiente línea de código podemos observar la forma en que se programaron las instrucciones para agregar una huella.

```
Serial.println("Listo para guardar huella!");
Serial.println("Elegir un número donde se guardara la huella");
id = readnumber();
if (id == 0) {
    return;
}
Serial.print("Guardando huella#");
Serial.println(id);

Serial.println("Quitar dedo");
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER) {
    p = finger.getImage();
}
Serial.print("ID "); Serial.println(id);
p = -1;
Serial.println("Colocar mismo dedo");
while (p != FINGERPRINT_OK) {
    p = finger.getImage();
    switch (p) {
        case FINGERPRINT_OK:
            Serial.println("Imagen tomada");
            break;
    }
}
Serial.print("ID "); Serial.println(id);
p = finger.storeModel(id);
if (p == FINGERPRINT_OK) {
    Serial.println("Guardada!");
}
```

Línea de código 6 Almacenamiento huella digital

Por cuestiones de seguridad se pensó que, para poder tener la posibilidad de agregar una huella nueva, sea únicamente con el proveedor. Por lo que la pantalla que se ve a continuación, solamente la vea el programador donde de igual forma las instrucciones son sencillas.

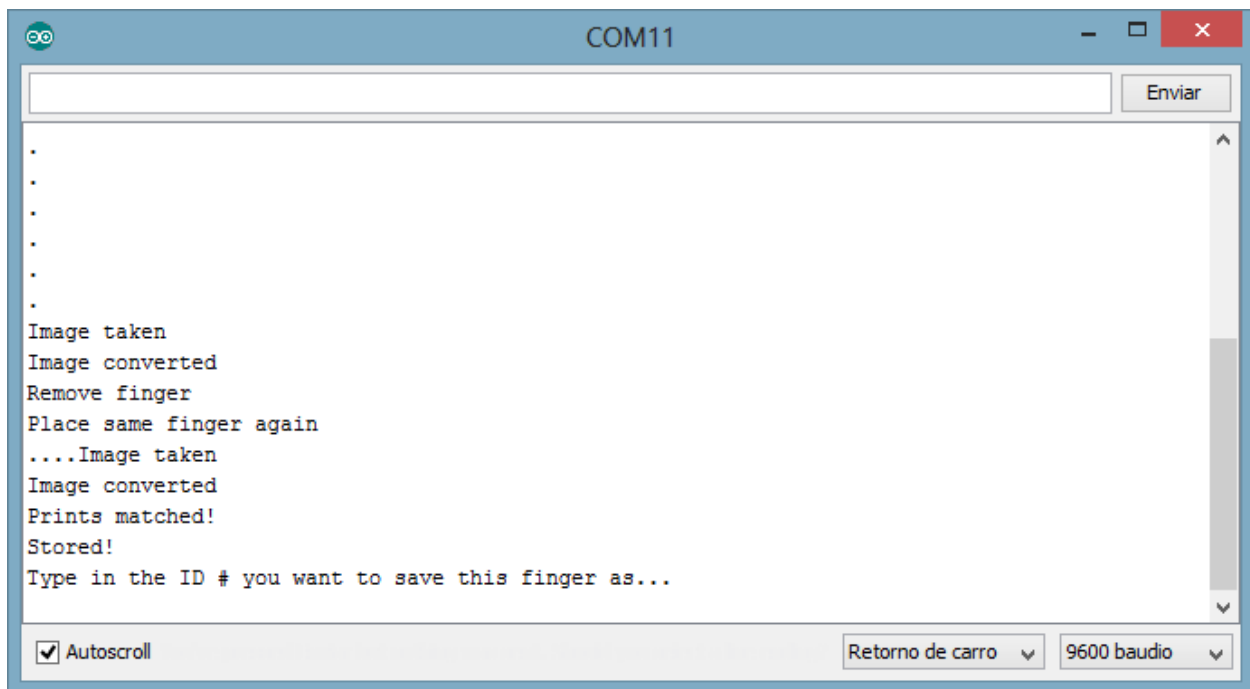


Figura 26 Prueba toma de imagen de huella digital

Una vez que se haya agregado la huella del usuario a la base de datos, ya se pueden iniciar con el encendido del carro y el ajuste de los espejos laterales. Para eso se necesitan seguir los pasos que se muestran en el display LCD, donde aquí ya no es necesaria la presencia del programador si no solamente el usuario que hará uso del automóvil cabe destacar que se aceptaran solamente el ingreso de 4 usuarios, aunque el sensor tenga la capacidad para almacenar más

“Bienvenido”

“Colocar dedo para arrancar auto”

Se borra la pantalla y ahora se muestra lo siguiente:

“Para guardar posición de los espejos apretar el botón”

Se borra la pantalla

“Posición guardada”

Como se puede observar las instrucciones son muy sencillas para la comodidad del usuario, ya que lo único que necesita hacer es mover el espejo con el controlador y después apretar el botón para guardar esa posición en la memoria del sensor.

Diagrama
Metología V

12.6 Display LCD

Las instrucciones que se mostraron anteriormente son las que debe de seguir el usuario, pero ¿Cómo va a ver las instrucciones?

Como se mencionó, será mediante un display LCD de 16x2 el cual nos permite mostrar 16 caracteres a la vez en 2 renglones y es de un tamaño reducido por lo que se puede colocar prácticamente en cualquier parte del automóvil.

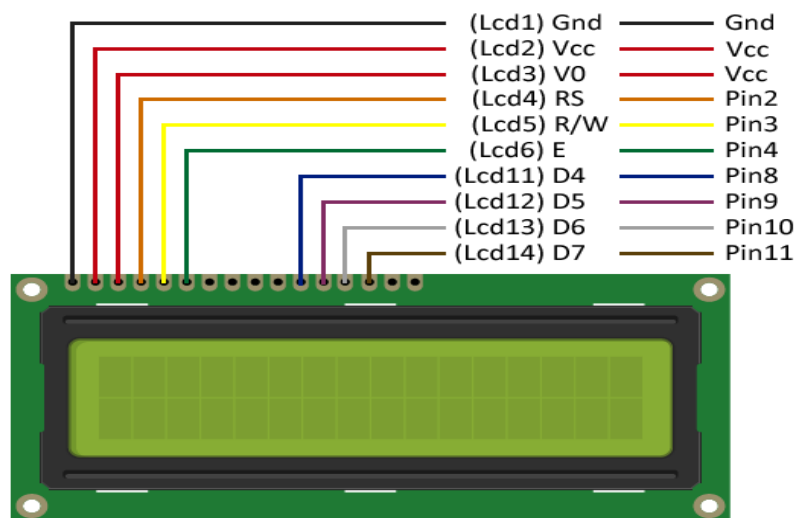


Figura 27 PinOut display LCD

Este tipo de display puede tener 3 formas distintas de comunicarse, la cual en esta ocasión se usó la directa, ya que tenemos varios sensores conectados mediante I²C.

La conexión de este display al Arduino es de la siguiente forma:

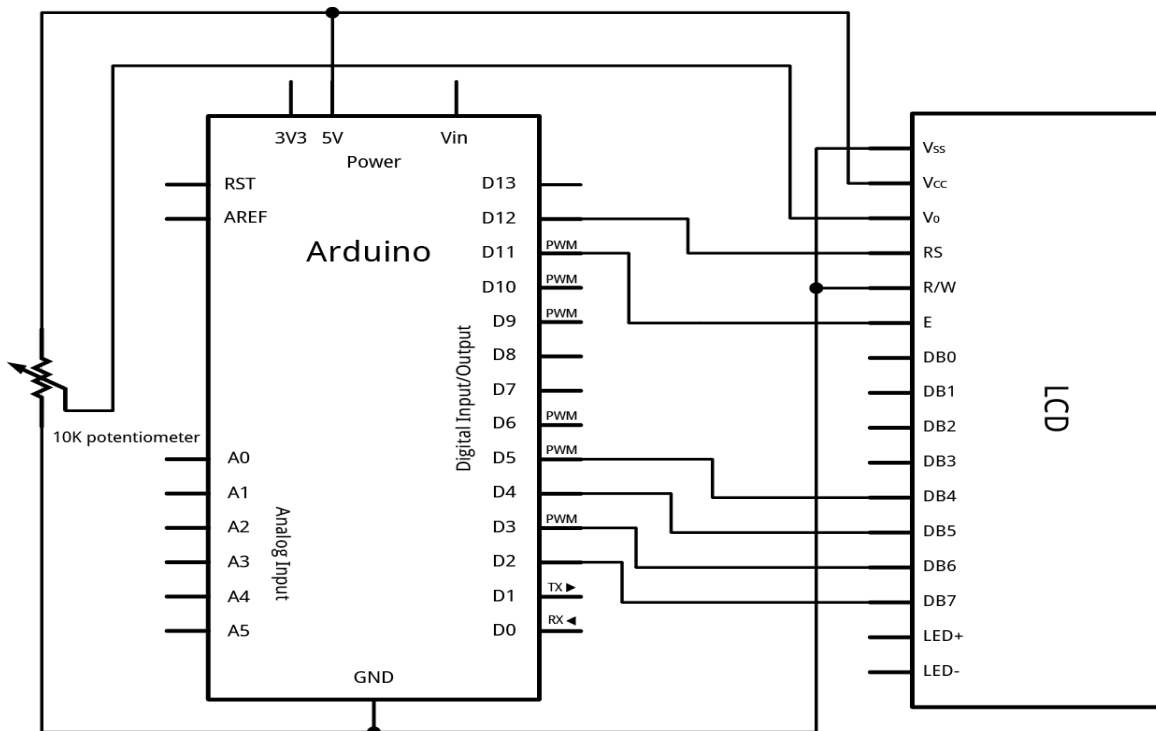


Figura 28 Conexión de display LCD a Arduino

Como podemos observar, se tiene un potenciómetro, este es para poder ajustar el contraste del LCD a forma de que se vea claro los caracteres. Una vez ajustado lo visual, se procede a la programación de los comandos, donde simplemente proyectaremos las instrucciones que debe seguir el usuario

```
Lcd.print("Bienvenido");
Lcd.clear();
Lcd.print("Colocar dedo");
Lcd.clear();
Lcd.print("Para guardar posición");
Lcd.setCursor(1, 8);
Lcd.print("espejos apretar botón");
Lcd.clear();
```

Línea de código 7 Display LCD

Las líneas de código para programar el LCD son sencillas, ya que mediante una librería usamos comandos predeterminados donde podemos hacer desde mostrar un simple texto, hasta eliminar la pantalla e iniciar el cursor en la posición que nosotros deseemos.

En esta ocasión solo se necesitaron los comandos de imprimir los cuales son `lcd.print(" ")` y borrar pantalla es `lcd.clear()`; se necesitaron solo de estos comandos ya que lo que se muestra en la pantalla son instrucciones, entonces conforme el usuario vaya realizando los pasos la pantalla va actualizando el paso en el que va el usuario, esto para no tener errores en el arranque del sistema.

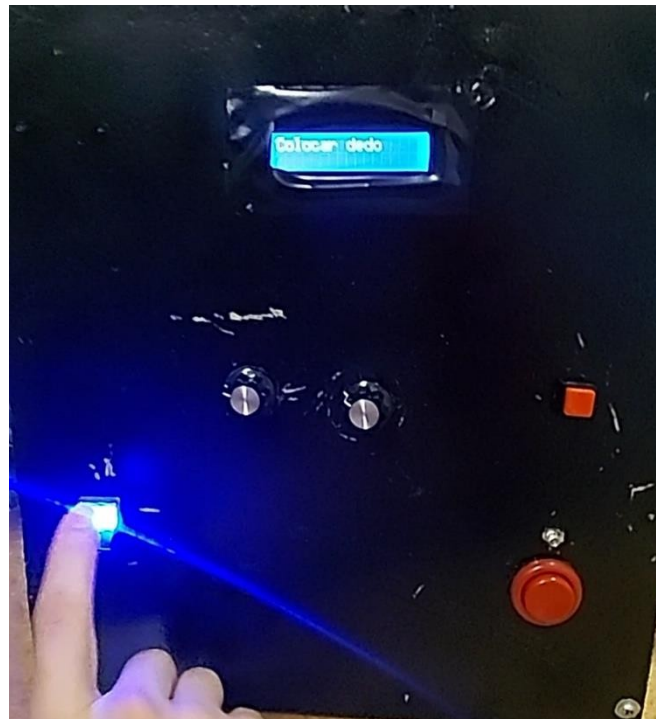


Figura 29 Visualización de instrucciones en display LCD

En la *figura 29* podemos ver la forma en que se observan las instrucciones para el usuario, donde se debe de colocar en una posición que se pueda voltear a ver de manera rápida y sencilla para que no se distraiga fácilmente.

12.7 Alimentación

El Arduino puede ser alimentado de varias formas, ya que tiene 3 entradas de alimentación, como se muestra en la *figura 30*.

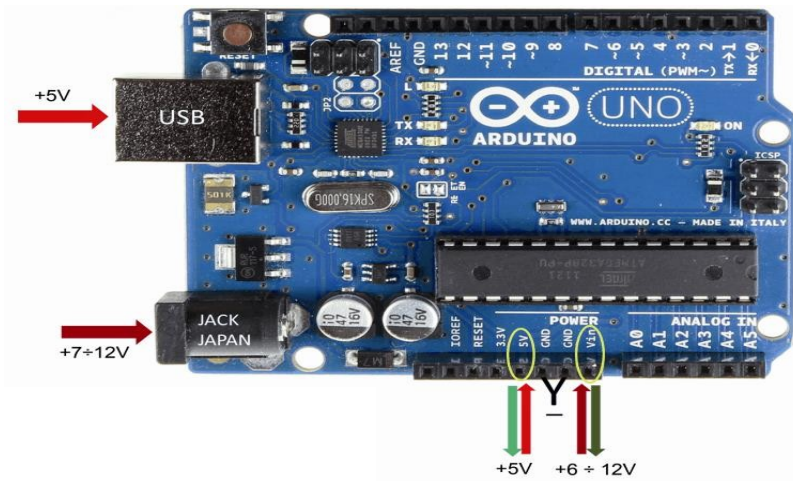


Figura 30 PinOut de alimentación de Arduino

La alimentación que se ocupará será un eliminador el cual puede variar su voltaje desde 3.3V hasta 12V, porque el voltaje que necesitan los motores es de 9V y entonces para poder ocupar una misma alimentación haremos un arreglo para poder ocupar el Jack del eliminador para alimentar tanto el Arduino como los motores.



Figura 31 Polaridad de conector jack

En la entrada Jack el positivo es que esta en la parte interna y el negativo es la parte exterior, como se muestra en la *figura 31*, por lo que se sacaron 2 cables y se conectaron al Arduino en el pin Vin, ya que ese soporta desde 6V hasta 12V, como se observa en la *figura 30* y otros 2 cables se sacaron para la

alimentación de los motores y se conectó el positivo al pin 8 del puente H, como se observa en la *figura 32* y el negativo al común.

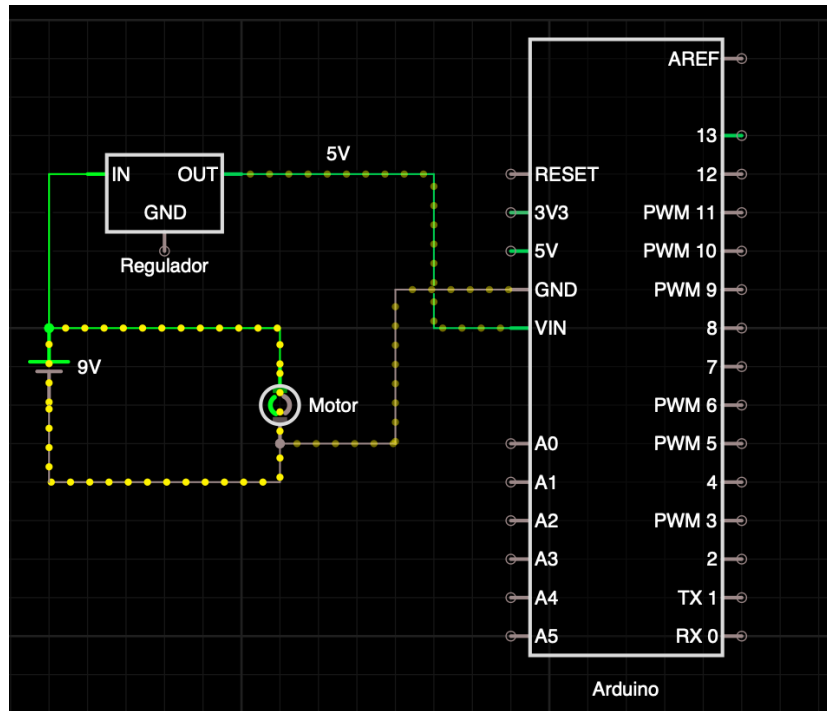


Figura 32 Alimentación de Arduino y Motores con misma fuente

En este diagrama podemos observar de forma general la conexión que se hizo para alimentar los motores y el Arduino con la misma fuente. Se separa la alimentación debido al consumo de corrientes de los sensores y los motores, ya que el Arduino no tiene la corriente suficiente para alimentar todo.

Revisando el datasheet de cada componente podemos obtener los siguientes datos:

- Motores DC consumen 500mA
- Sensor MPU6060 consume 3.9mA
- Lector de huella digital consume 150mA
- Puente H consume 600mA

La ley segunda Ley de Kirchoff nos dice que todas las corrientes que entran en un nudo debe ser iguales a las que salen, por lo tanto, tenemos que:

$$\sum_{k=1}^n I_k = I_1 + I_2 + I_3 + \dots + I_n$$

$$\sum 500mA + 3.9mA + 65mA + 600mA = 1.253A$$

Es decir, se necesita una fuente que nos proporcione más de 1.253A y nosotros solo tenemos eliminadores que nos dan hasta 1200mA, por lo que necesitaremos al menos 2 fuentes para alimentar todos los componentes

12.8 Integración de todos los componentes

El desarrollo del proyecto se fue haciendo por pasos para tener un orden y poder integrar todo de forma adecuada.

Se empezó con el control de los espejos, el cual engloba el uso del sensor MPU6050 y el puente H lm293D, donde también se hizo un control manual de los espejos para que el usuario lo moviera en la posición que el desee.

Posteriormente se hizo la programación del sensor de huella digital donde se integró el control de los espejos para que se muevan de forma automática.

Una vez teniendo todo el programa integrado, los espejos y sensores se colocaron en el banco de prueba.

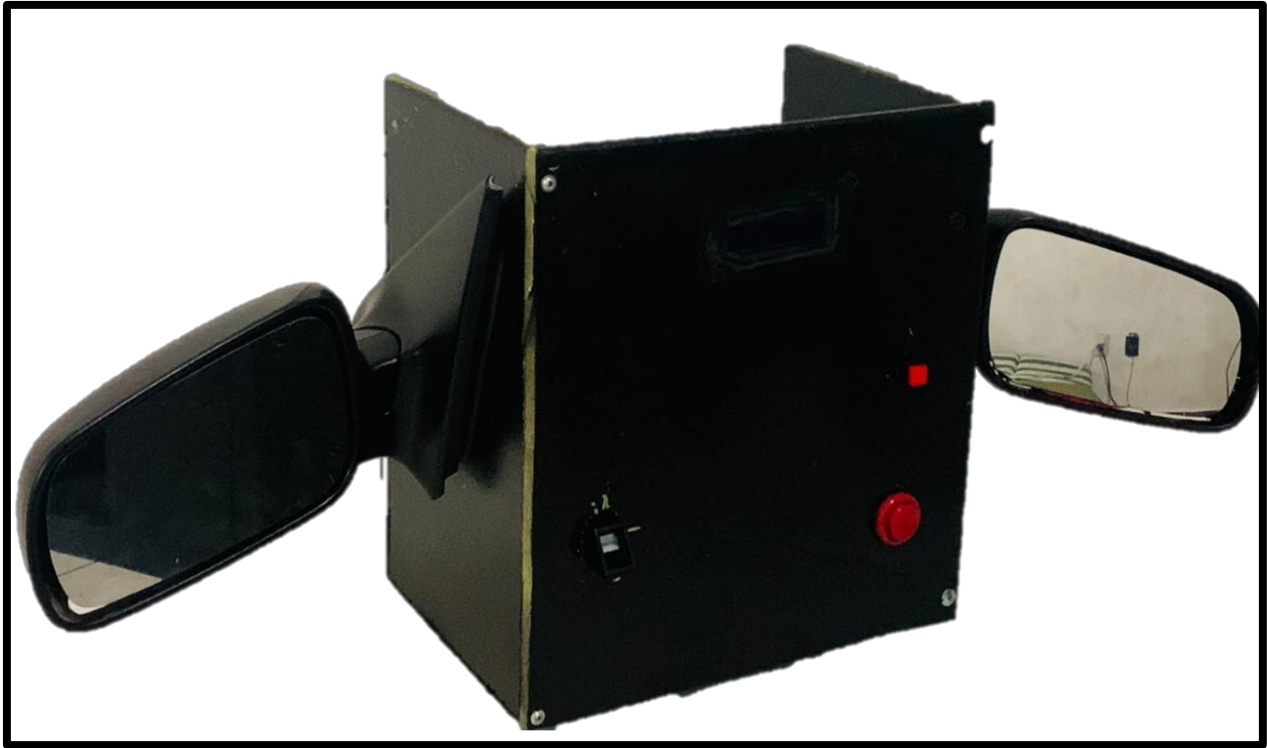


Figura 33 Banco de prueba

En la *figura 33* podemos observar la forma en que quedo todo implementado en el banco de pruebas, la forma en que quedo distribuido es para que el usuario pueda observar el movimiento de los espejos y las instrucciones en la pantalla. Este banco se tuvo que hacer derivado de la pandemia ya que no se tenía la accesibilidad del carro, el material del que esta hecho se llama *Durawood* es rígido, pero a la vez es maleable, esto me permitió poder hacer los barrenos de los sensores y la pantalla de forma sencilla.

A continuación, se darán los pasos para poder arrancar el sistema:

- 1) Conectar la alimentación.
- 2) Revisar que el sensor de huella y la pantalla estén encendidas
- 3) Leer instrucción en la pantalla.
- 4) Mover espejos en la posición deseada.
- 5) Apretar botón para guardar posición.
- 6) Apretar botón para encender carro una vez que se haya colocado la huella.

Al seguir estas instrucciones el sistema deberá de funcionar de forma correcta. En la *figura 34* observamos los componentes que están en el banco para poder identificarlos:

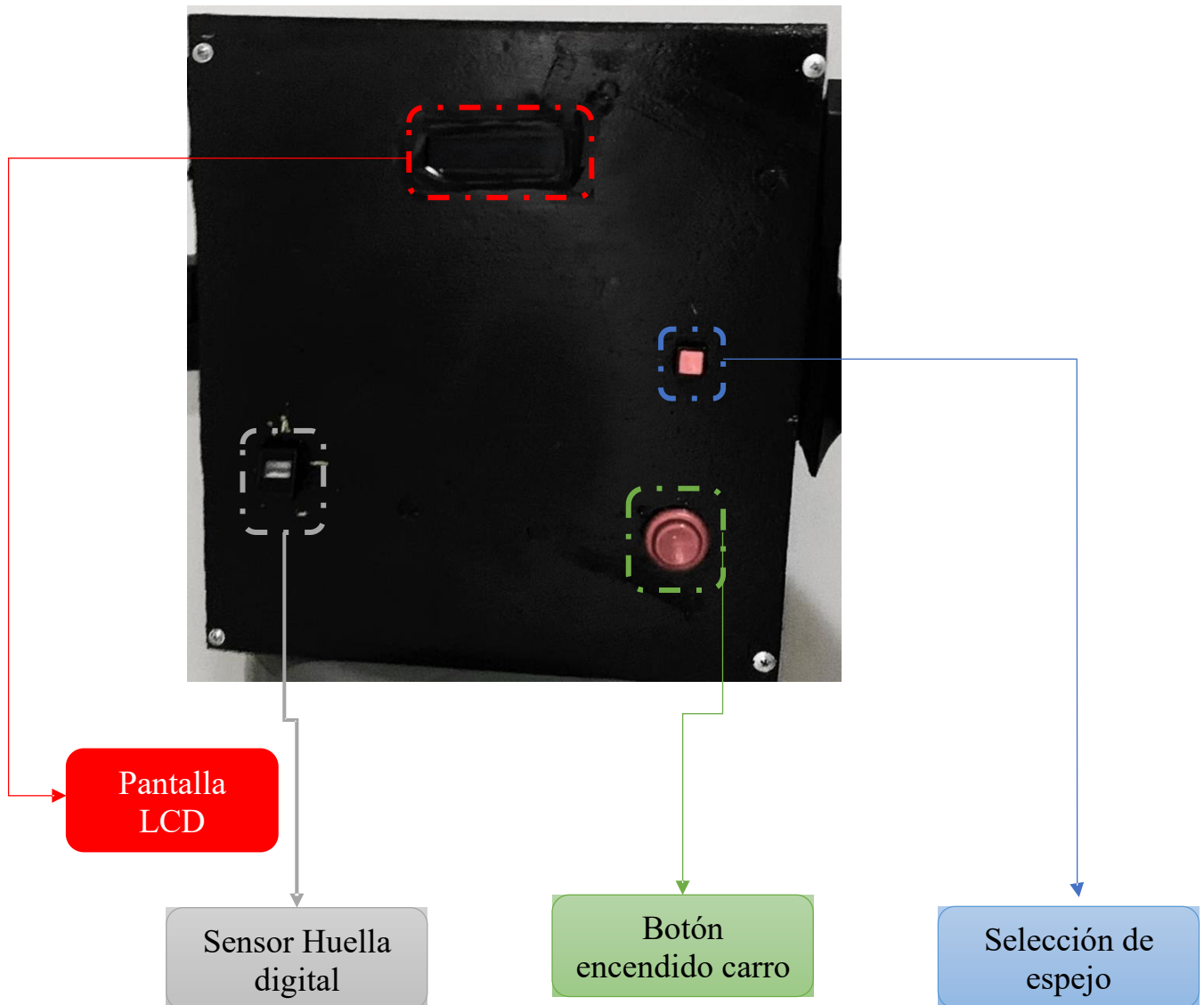


Figura 34 Componentes del banco de prueba

13. Conclusiones

El proceso que se llevó a cabo para poder desarrollar este proyecto tuvo ciertas complicaciones debido a el acondicionamiento de los sensores y la estandarización, esto porque el sensor MPU6050 es uno de los que necesita mayor precisión y definir los parámetros que se ocuparan mediante programación. Al final se logro el objetivo, el cual era crear una forma de que el usuario sintiera mas seguridad en su automóvil y a la vez tener mejor confort con el simple toque de un dedo.

El aprendizaje que se tuvo de estas complicaciones es la importancia de conocer el funcionamiento de dicho sensor y saber leer la hoja de datos, esto para aprovechar al máximo el sensor desde el gasto energético, hasta la lectura de datos.

También por la situación que está atravesando el mundo, se tuvo que rediseñar el sistema que se tenía planteado, ya que no se fue posible hacer la instalación en el automóvil, por lo que se necesitó pensar de manera rápida una alternativa la cual fuera funcional para mostrar el funcionamiento de todo el sistema de manera clara.

14. Trabajo a futuro

Diagrama
Metología V

Con el desarrollo de este proyecto se puede seguir mejorando el confort del conductor en un automóvil, es decir, agregar sistemas de confort al momento de querer encender el automóvil con el mismo sistema de identificación de usuario. Es por eso la importancia de la mejora continua para tener como resultado un proyecto que este completamente desarrollado y se cubran todas las necesidades de las personas.

Algunos de los sistemas que se pueden agregar para mayor confort son:

1. Ajuste de altura y profundidad del volante
2. Ajuste de asiento de conductor
3. Ajuste de espejo retrovisor

15. Referencias

- 1) *Aprendiendo Arduino* Obtenido de:
<https://aprendiendoarduino.wordpress.com/2016/09/25/que-es-arduino/>
- 2) Celedonio Enrique Aguilar Meza, I. R. (s.f.). *Boletín UPIITA*. Obtenido de:
<http://www.boletin.upiita.ipn.mx/index.php/ciencia/205-cyt-numero-23/644-implementacion-de-la-huella-digital-para-el-encendido-de-un-automovil>
- 3) Coello Torres, P. E. (2014). "Diseño de un sistema de encendido automático e inmovilizador para un vehículo estándar, por medio de la adaptación de un dispositivo lector de huellas digitales." Ecuador, Riobamba.
- 4) Gabriel Sanchez, K. T. (2008). *Universidad de Antioquia*. Obtenido de Revista Facultad de Ingeniería:
<http://aprendeenlinea.udea.edu.co/revistas/index.php/ingenieria/article/view/17933/15402>
- 5) HETPRO. *HETPRO store*. Obtenido de <https://hetpro-store.com/lector-de-huella-digital-071405/>
- 6) INEGI. (2017). *INEGI*. Obtenido de <http://www.beta.inegi.org.mx/temas/vehiculos/>
- 7) *Inngresa*. (2016). Obtenido de <https://inngresa.com/lectores-biometricos-huella-digital-mayor-seguridad/>
- 8) Peralta, (22 de Noviembre de 2017). La importancia del uso de los espejos retrovisores. *El universal* .
- 9) Político,(2018). *Animal Político* Obtenido de:
<https://www.animalpolitico.com/2018/01/autos-robo-record-2017/>
- 10) Felipe, V. L. B. Sensores electrónicos como remplazo del sentido auditivo.
- 11) <https://inversense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>
- 12) <https://es.aliexpress.com/item/32590598701.html>
- 13) Pérez, E. M. (2007). Microcontroladores PIC: sistema integrado para el autoaprendizaje. Marcombo.

16. Anexos

16.1 Hoja de especificaciones sensor MPU6050

6 Electrical Characteristics

6.1 Gyroscope Specifications

VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T_A = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
GYROSCOPE SENSITIVITY						
Full-Scale Range	FS_SEL=0 FS_SEL=1 FS_SEL=2 FS_SEL=3		±250 ±500 ±1000 ±2000		%/s %/s %/s %/s	
Gyroscope ADC Word Length			16		bits	
Sensitivity Scale Factor	FS_SEL=0 FS_SEL=1 FS_SEL=2 FS_SEL=3		131 65.5 32.8 16.4		LSB/(°/s) LSB/(°/s) LSB/(°/s) LSB/(°/s)	
Sensitivity Scale Factor Tolerance	25°C	-3		+3	%	
Sensitivity Scale Factor Variation Over Temperature			±2		%	
Nonlinearity	Best fit straight line; 25°C		0.2		%	
Cross-Axis Sensitivity			±2		%	
GYROSCOPE ZERO-RATE OUTPUT (ZRO)						
Initial ZRO Tolerance	25°C		±20		%/s	
ZRO Variation Over Temperature	-40°C to +85°C		±20		%/s	
Power-Supply Sensitivity (1-10Hz)	Sine wave, 100mVpp; VDD=2.5V		0.2		%/s	
Power-Supply Sensitivity (10 - 250Hz)	Sine wave, 100mVpp; VDD=2.5V		0.2		%/s	
Power-Supply Sensitivity (250Hz - 100kHz)	Sine wave, 100mVpp; VDD=2.5V		4		%/s	
Linear Acceleration Sensitivity	Static		0.1		%/s/g	
SELF-TEST RESPONSE						
Relative	Change from factory trim	-14		14	%	1
GYROSCOPE NOISE PERFORMANCE	FS_SEL=0					
Total RMS Noise	DLPFCFG=2 (100Hz)		0.05		%/s-rms	
Low-frequency RMS noise	Bandwidth 1Hz to 10Hz		0.033		%/s-rms	
Rate Noise Spectral Density	At 10Hz		0.005		%/s/√Hz	
GYROSCOPE MECHANICAL FREQUENCIES						
X-Axis		30	33	36	kHz	
Y-Axis		27	30	33	kHz	
Z-Axis		24	27	30	kHz	
LOW PASS FILTER RESPONSE						
	Programmable Range	5		256	Hz	
OUTPUT DATA RATE						
	Programmable	4		8,000	Hz	
GYROSCOPE START-UP TIME						
ZRO Settling (from power-on)	DLPFCFG=0 to ±1%/s of Final		30		ms	

1. Please refer to the following document for further information on Self-Test: *MPU-6000/MPU-6050 Register Map and Descriptions*

16.2 Hoja de especificaciones sensor huella digital

Power	DC 3.6V-6.0V	Interface	UART(TTL logical level)/ USB 1.1
Working current	Typical: 100mA Peak: 150mA	Matching Mode	1:1 and 1:N
Baud rate	(9600*N)bps, N=1~12 (default N=6)	Character file size	256 bytes
Image acquiring time	<1s	Template size	512 bytes
Storage capacity	120/ 375/ 880	Security level	5 (1, 2, 3, 4, 5(highest))
FAR	<0.001%	FRR	<0.1%
Average searching time	< 1s (1:880)	Window dimension	14mm*18mm
Working environment	Temp: -10°C - +40°C	Storage environment	Temp: -40°C - +85°C
	RH: 40%-85%		RH: <85%
Outline Dimention	Split type	Module: 42*25*8.5mm (install dimension: 31.5*19mm) Sensor:56*20*21.5mm	
	Integral type	56*20*21.5mm	

16.3 Hoja de especificaciones potenciómetro



PTT-B1K - 16 mm Rotary Potentiometer

Features:

- Long Life / Low Cost
- Knurled Shaft
- Panel Mount
- Metal Shaft
- PC Pins PCB
- Linear Taper
- Breadboard Friendly

Electrical Characteristics

Taper.....Linear
 Standard Resistance.....0 to 1K ohms
 Standard Resistance Tolerance.....±20 %
 Terminal Resistance.....10 ohms max.

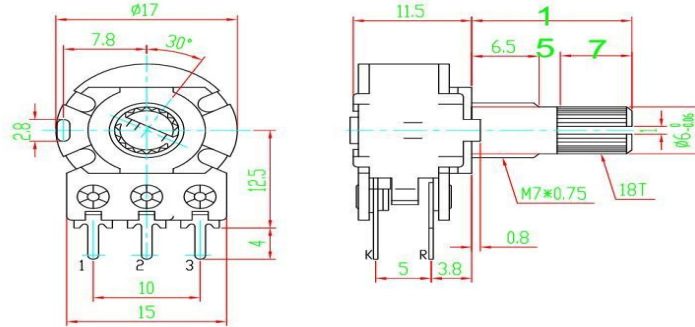
Environmental Characteristics

Operating Temperature.....-10 °C to +75 °C
 Power Rating0.125 watt
 Maximum Operating Voltage200 V
 Rotational Noise68 mV max.

Mechanical Characteristics

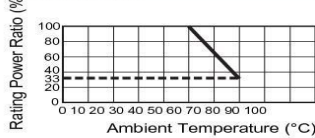
Mechanical Angle300 ° ±5°
 Rotational Torque10 to 150 g-cm
 Stop Strength.....4 kg-cm min.
 Rotational Life.....10,000 cycles min.
 Electrical Termination.....PCB Mount
 HardwareWashers and Nuts

Product Dimensions

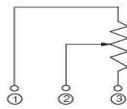


All Dimensions in millimeter unless specified otherwise

Derating Curve



Circuit



How To Order

Model.....PTT-B1K

16.4 Hoja de especificaciones puente HLM293D

L293D - L293DD

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_s	Supply Voltage	36	V
V_{ss}	Logic Supply Voltage	36	V
V_i	Input Voltage	7	V
V_{en}	Enable Voltage	7	V
I_o	Peak Output Current (100 μ s non repetitive)	1.2	A
P_{tot}	Total Power Dissipation at $T_{pin}=90$ ° C 4		W
T_{stg}, T_j	Storage and Junction Temperature	- 40 to 150	° C

PIN CONNECTIONS (Top view)

