

**Instituto Tecnológico y de Estudios Superiores
de Monterrey**

“Campus Monterrey”



**TC1032.207 - Modelación de sistemas mínimos y
arquitecturas computacionales (Gpo 207)**

Mtro. Abraham Jashiel Pérez Estrada

“Situación problema”

- Reporte Técnico Final -

Marco Antonio Rodríguez Amezcua A00834672

David Eugenio Cavazos Wolberg A01721909

Roberto González Reyes A00833852

Rodolfo Charles Wah A01383393

Yuvan Thirukumaran A00834121

22 Octubre 2022

Objetivo del proyecto:

Resolver la problemática que se genera cuando uno de los faros principales en el automóvil, primordialmente los traseros, deja de funcionar y no se le da una notificación de ello al conductor, utilizando el tablero de instrumentos.

Un programa manda constantemente 10 datos de las corrientes eléctricas generadas por los 2 focos traseros del automóvil (cada uno con un sensor para medir el voltaje). Así, cada cierto tiempo se saca la media de ese voltaje y se manda al tablero (o a otro lugar, aún falta por definir esa parte) dependiendo del sensor del foco que se elija.

Introducción e identificación de la problemática automotriz:

La iluminación externa del automóvil es una de las herramientas principales al momento de estar en la calle entre una multitud a gran escala de otros vehículos. Esto es debido a que gracias a estos dispositivos lumínicos podemos visualizar de mejor manera nuestro trayecto, y así evitar cualquier accidente, los faros también nos ayudan a comunicarnos con los demás autos a nuestro alrededor, nos ayudan a avisarle a los demás cuando reducimos nuestra velocidad, cuando vamos a girar, etc. En relación a estos puntos, es muy importante que las luces traseras sigan funcionando en todo momento ya que al no funcionar adecuadamente, los conductores que nos rodean no saben lo que estamos haciendo, lo cual pudiera ocasionar un accidente vial de alto riesgo.

En varios países, las inspecciones técnicas realizadas a los automóviles en circulación han presentado algunos casos de inconvenientes con su sistema lumínico, en el apartado de alumbrado y señalización. Un ejemplo es el índice de inspecciones técnicas realizadas por el Ministerio de Industria, en España, del cual el 19% fueron negadas para impedir la circulación de varios vehículos. Tomando como referencia esto, debemos tener en cuenta la constante revisión de nuestros instrumentos electrónicos para un funcionamiento óptimo de nuestros vehículos.

Identificación de variables en la problemática:

Para que la persona que conduce el coche reciba una notificación cuando alguna luz trasera no funciona correctamente, podemos utilizar un sistema de alerta que se muestre en el tablero de instrumentos, para ver la indicación y si las luces traseras funcionan correctamente o tienen algún tipo de fallo. Para implementar este tipo de cosas, hay que tener en cuenta diferentes sensores y módulos que tenemos a nuestra disposición. Estos sensores pueden detectar alguna ausencia de corriente o voltaje al tratar de prender los faros traseros. Estos diferentes sensores se deben implementar de tal forma que sean visibles para el conductor en todo momento.

Sabemos que todos los coches modernos vienen con tres colores diferentes que indican una luz de advertencia, rojo, ámbar y verde. Las luces de advertencia rojas significan que hay que parar el coche en cuanto sea seguro ya que hay una falla

grave en el mismo. Las luces de advertencia amarillas significan que es necesario actuar lo más pronto posible, pero no es tan grave como un foco rojo. Las luces de advertencia verdes son sólo informativas y usualmente no indican ningún problema en específico. Para poder aplicar la solución de nuestro problema, podemos aprovechar la modernidad de los coches que manejamos hoy en día, la mayoría de ellos están equipados con sensores para controlar su estado, es por eso que creemos que no sería muy difícil implementar nuestra solución. Se puede trabajar con algún sensor que nos ayude a regular la calidad de los faros traseros del auto y nos puedan indicar cuando los faros parecen no funcionar correctamente.

Así mismo, debemos de tomar en cuenta todas las variables que estén involucradas en el mal funcionamiento de este sistema. Esto va desde un fusible quemado, lo cual no permite la notificación o alerta de esta problemática, hasta detectar que la bombilla se encuentra fundida. Puede que en algunas situaciones el origen de la problemática varíe debido a que, como se mencionó anteriormente, todo puede deberse a la instalación errónea de un aparato electrónico o que su durabilidad le impida seguir funcionando correctamente.

Se tienen en cuenta muchas variables a la hora de implementar un sensor de este tipo, como el calcular bien los números, es importante considerar nuestro tiempo para construirlo ya que los sensores serán fundamentales para desbloquear altos niveles de automatización, y se espera que el número y el tipo de sensores aumente. Otra variable puede ser asegurar la ubicación correcta, deben estar colocados estratégicamente para retroalimentar la información sobre el entorno del coche de forma continua, pero tomando en cuenta que no es algo vital para el coche, por lo tanto no debe estorbar otros anuncios más importantes. Además, hay que asegurarse de no sobrecargar el coche implementando más y más sensores para alertar al conductor. Puede ser arriesgado que el conductor se distraiga con todas las funcionalidades y se pierda información vital para la conducción. Variables como estas son importantes y hay que pensar en ello en todo momento durante la implementación.

Asimismo, hay muchas ventajas y desventajas que vienen con esta solución, una de las ventajas de esto es que estos sensores pueden hacer la vida del conductor más fácil, se hacen para asegurar que no se cause ningún daño al coche. Asegurando que el conductor pueda conocer cualquier daño que se produzca en el coche mediante un tell-tale en el tablero de instrumentos. La desventaja que viene con esto es que tienen tantos sensores a bordo que pueden fallar con el tiempo. Un sensor defectuoso puede provocar aún más daños al coche.

Los faros de un automóvil son una de las herramientas más fundamentales y necesarias para la seguridad del conductor, los pasajeros del mismo carro y de la demás gente en la calle. Cuando la luz trasera de un vehículo no funciona es muchas veces difícil que el conductor se de cuenta por sí solo, a no ser que le avisen o que él mismo vea su carro desde afuera. Es por esta razón, que creemos

necesario implementar un sensor en el automóvil que notifique al conductor cuando las luces traseras no estén funcionando correctamente. Para saber qué sensor usar es importante considerar todas las maneras en las que un faro trasero puede fallar, esto puede ser desde un fusible quemado hasta una bombilla fundida, y a partir de esto ya podemos determinar cómo utilizar nuestros sensores y como ponerlos en el tablero de instrumentos. Como todos sabemos, la gran mayoría de los carros que vemos hoy en día, tienen sensores que advierten al conductor si algo está fallando en su motor o en sus llantas. Es por esta razón que creemos que es posible implementar esta solución y es algo necesario ya que es preferible que el conductor se de cuenta que sus focos traseros no funcionan a través de un sensor, a esperar a que un accidente suceda.

Codificación de nuestra solución

Para lograr implementar nuestra solución hacía esta problemática que estamos enfrentando, desarrollamos un programa que recibe constantemente datos de las corrientes eléctricas generadas por los 2 focos traseros del automóvil (cada uno con un sensor para medir el voltaje). Así, cada cierto tiempo se saca la media de ese voltaje y si esta media es más baja que lo que debería de ser, una señal se manda al tablero de instrumentos y se despliega la luz de advertencia del tell-tale.

Tomando como referencia un sensor que entrega una lectura a través de un bus paralelo a razón de 12 bits/ms, realizamos un programa de MARIE, que almacena las primeras 20 mediciones en memoria.

Construimos un código que utiliza la función StoreI para ir guardando los valores

```
1  loop, Input
2      Store a
3      Load a
4      StoreI b
5      Load b
6      Add one
7      Store b
8      Load c
9      Subt one
10     Store c
11     Skipcond 800
12     Halt
13  Jump loop
14
15  a, DEC 0
16  b, DEC 0
17  c, DEC 20
18  one, DEC 1
```

resultantes en las primeras casillas (debido a que al inicializar la variable el StoreI toma el cero que la representa en ese momento para ubicar el nuevo valor), para posteriormente cambiar el número de casilla para el próximo valor (con Add). Así

mismo, decrece por cada ciclo un contador que nos marcará el número de datos totales tomados para así dar un límite de 20 mediciones al programa.

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
000	0000	0000	0001	0005	0006	0005	0004	0006	0008	0010	0015	0020	0021	0019	0015	0012
010	0006	0006	0005	0003	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
020	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
030	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
040	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
050	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
060	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

Podemos observar cómo se guardan las mediciones en los primeros registros consecutivamente.

De igual manera, aquí se presentan los registros que se utilizaron para el proceso e instrucciones del programa.

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0E0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0F0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
100	5000	210D	110D	E10E	110E	3110	210E	110F	4110	210F	8800	7000	9100	0004	0014	0000
110	0001	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
120	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
130	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
140	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

Para calcular el tamaño del programa tenemos que sumar la cantidad de registros totales y multiplicar por dos, ya que un registro es equivalente a dos bytes. Podemos representarlo de la siguiente manera:

$$20 (\text{mediciones}) + 17 (\text{instrucciones}) = 37 \rightarrow 37 * 2 = 74 \text{ bytes}$$

Por lo tanto, el tamaño de este programa es de **74 bytes**.

Y como parte de nuestro trabajo final, realizamos un programa que después de guardar en memoria los datos leídos del sensor, obtenga la media estadística de las mediciones almacenadas.

1) Entrada y almacenamiento de datos:

Assembly code:

```
1  ORG 100
2
3  loop, Input
4      Store a
5      Load a
6      StoreI b
7      Load b
8      Add one
9      Store b
10     Load cont
11     subt one
12     Store cont
13     Skipcond 800
14     Jump Sens2
15 Jump loop
16
```

Assembly code:

```
21 Sens2, Input
22     StoreI PosSens2
23     Load PosSens2
24     Add one
25     Store PosSens2
26     Load cont2
27     subt one
28     Store cont2
29     Skipcond 400
30     Jump Sens2
31
32 Load PosSens2
33 subt one
34 Store PosSens2
35
36 / Loop total suma
37 TSuma, Load dividendo
38     AddI PosSens2
```

Almacenamiento de datos dentro de una variable predeterminada para cada sensor (1 y 2). Almacenamiento de datos dentro de otra variable que se incrementará en 1 para cambiar la posición de la tienda de forma consecutiva (StoreI). Decrementa el contador de cada sensor después de cada nueva entrada. El estado de los contadores se evalúa con la función "Skipcond".

2) Suma de valores del sensor:

Assembly code:

```
32 Load PosSens2
33 subt one
34 Store PosSens2
35
36 / Loop total suma
37 TSuma, Load dividendo
38     AddI PosSens2
39     Store dividendo
40     Output
41     Load PosSens2
42     subt one
43     store PosSens2
44     subt two
45     Skipcond 400
46     Jump TSuma
47
48
49 / Division
```

Definimos un dividendo como un contador de los números totales a sumar. Luego sumamos el último valor obtenido del sensor 2. Vamos a la dirección de memoria anterior y restamos la posición de memoria del sensor 1 para comprobar si ya hemos sumado todos los valores del sensor 2.

3) División para sacar la media:

Assembly code:	Assembly code:
39 Store dividendo	58 regla1, Load cociente
40 Output	59 Add one
41 Load PosSens2	60 Store cociente
42 sub1 one	61 Jump restar
43 store PosSens2	62
44 sub1 two	63 regla2, Load cociente
45 Skipcond 400	64 Output /imprime el cociente
46 Jump TSuma	65 Halt
47	66
48	67 /SumTotal deberia ser 155
49 / Division	68 /Cociente debe ser 15
50 /Loop para restar	69
51 restar, Load dividendo	70 a, DEC 0
52 Sub1 divisor	71 b, DEC 0
53 Store dividendo	72 c, DEC 20
54 Skipcond 000	73 cont, DEC 10
55 Jump regla1	74 cont2, DEC 10
56 Jump regla2	75 PosSens2, HEX 010

Una vez que tenemos la entrada que se almacenan en los datos en este caso la localidad y suman de valores del sensor para hacer división. Tenemos que encontrar la media del segundo sensor. Como hay dos sensores, cada uno contiene 10 datos diferentes, y encuentra la media de salida del segundo sensor en Marie. Para encontrar la media, hemos implementado el método de división usando un loop para restar e imprimir el cociente.

4) Resultado:

The screenshot displays the Marie simulator interface. On the left, the 'Assembly code' window shows a loop for division. The 'Registers' window on the right shows the current state of the machine's registers. The 'Memory dump' at the bottom shows the contents of memory locations.

Assembly code:

```
1 ORG 100
2
3 loop, Input
4   Store a
5   Load a
6   StoreI b
7   Load b
8   Add one
9   Store b
10  Load c
11  Subt one
12  Store c
13  Load cont
14  sub1 one
15  Store cont
16  Skipcond 800
17  Jump Sens2
18 Jump loop
```

Registers:

Register	Value
AC	000F
IR	7000
MAR	133
MBR	7000
PC	134
IN	0014
OUT	000F

Memory dump:

Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
000	0001	0002	0003	0004	0005	0006	0007	0008	0009	000A	0000	0000	0000	0000	0000	0000
010	000B	000C	000D	000E	000F	0010	0011	0012	0013	0014	0000	0000	0000	0000	0000	0000
020	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
030	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
040	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
050	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
060	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
070	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

Para almacenar los resultados de cada sensor utilizamos un formato de 16 bits los cuales van del 0 al 15. En los primeros 12 bits, del 0 al 11, almacenamos los datos que nuestro sensor está leyendo, después, el bit 12 es un bit en blanco que se utiliza como espaciador, y los bits 13, 14 y 15 almacenan la media del sensor 1, la media del sensor 2, y la selección del usuario respectivamente.

En los resultados, podemos ver que al montar la localidad y ejecutarla nos pide las 10 instrucciones del primer sensor y las 10 siguientes para el segundo. Una vez que hemos introducido la instrucción en su localidad designada, comenzamos a ejecutar el código y a encontrar la media del sensor dos que se mostrará en el modo de salida.

Estas 10 instrucciones que estamos introduciendo para cada sensor es la tensión de la luz del coche. El primer sensor está conectado al faro delantero y el segundo sensor está conectado a los faros traseros. Así, siempre que la luz deje de funcionar, estará leyendo constantemente los datos de las corrientes eléctricas generadas por los 2 faros traseros del coche y los mostrará en el salpicadero del coche indicando si los faros funcionan o no. Antes, hemos investigado un poco la tensión de las luces del coche para saber qué rango de tensión debemos introducir para que el ensamblador lea y muestre en el modo de salida. La tensión de alimentación de entrada de tiempo entre la parada o la cola y la tierra es nominal de 6V a 16V, (precisamente 13.2) y hasta 45V durante un volcado de carga.

Por lo tanto, como valores de referencia para saber si los focos están fallando o no, tomaremos que si el promedio de las mediciones está por debajo de los 12.5 volts por un periodo de tiempo prolongado, ahí es cuando se le avisará al usuario en el tablero que una de sus luces traseras está fallando.

Reflexión:

Realmente la tecnología ha innovado y revolucionado muchas de nuestras tareas laborales y domésticas a lo largo de nuestra vida cotidiana. Esto se remonta desde haber conseguido escribir, recibir y calcular un mensaje en una máquina que utilizaba tubos al vacío y tenía problemas con su memoria, hasta las computadoras actuales que recopilan datos a través de sensores y sistematizan procesos. Esto fue de gran ayuda en diferentes industrias, que tienen bastante enfoque en la producción en serie, como lo es la automotriz, la telefónica, entre otras. Además, estas herramientas son la principal fuente de trabajo e información para la generación y mantenimiento de las redes. Sabemos que antes los carros no tenían ni un sensor en ellos, hasta que un día se les agregaron dos sensores, uno para el aceite y otro para la batería, ya hoy en día podemos encontrar más de 70 sensores en un carro. Aunque esta cantidad de sensores parezca más que suficiente, si están cubriendo una necesidad real que tiene el consumidor de saber qué está pasando con su carro ya que este tiene muchos mecanismos que aseguran el bienestar del conductor. Agregar sensores que detectan cuando un faro trasero tenga que ser reemplazado es un paso más para seguir cumpliendo con esta meta.

Este es un problema en general que se encuentra en todas las áreas de la ingeniería, muy probablemente el área de la aeroespacial sufra esto en otro nivel, principalmente ya que una falla en un avión es mucho más grave que una falla en

un carro, es por eso que los aviones están repletos de todo tipo de sensores, y podemos aprender de ellos como se comunican entre sus microcontroladores y sus sensores. Y ya para terminar podemos decir que con lo mucho que ha evolucionado la tecnología en los últimos años, se nos hace muy posible que nuestra solución a esta problemática sea implementada, y después de codificar la solución en un programa como MARIE aún más. Realmente consideramos que de aplicarse se pudieran reducir los accidentes automovilísticos de manera significativa, a su vez, nuestra solución generaría una mayor concientización de la importancia de los faros en la sociedad.

Referencias:

Los ojos y Oídos del Coche - ¡Alrededor de los sensores! (n.d.). Retrieved September 23, 2022, from <https://www.expertoautorecambios.es/magazine/sensores-2494>

Microcontrolador - Qué es y para qué sirve. (2021, March 7). Retrieved September 23, 2022, from <https://hetpro-store.com/TUTORIALES/microcontrolador/>

Problemas con las luces de posición. Auto10.com | Expertos en coches. (n.d.). Retrieved September 23, 2022, from <https://www.auto10.com/reportajes/problemas-con-las-luces-de-posicion/477>

The critical 1.5 volts - high on-board voltage, short lamp life. OSRAM Carlight Blog. (2016, September 12). Retrieved October 22, 2022, from <https://www.carlightblog.com/2012/08/16/the-critical-1-5-volts-high-on-board-voltage-short-lamp-life/>

False. (n.d.). *LED-driver design for Automotive Rear Lig: Maxim integrated.* LED-Driver Design for Automotive Rear Lig | Maxim Integrated. Retrieved October 22, 2022, from <https://www.maximintegrated.com/en/design/technical-documents/app-notes/4/4316.html>