Instituto Tecnológico y de Estudios Superiores de Monterrey

Actividad. Avance 3 – Reto (Final)

TC1032.5 Modelación de sistemas mínimos y arquitecturas computacionales

Nombre del Profesor: Claudia M. Solís Garza

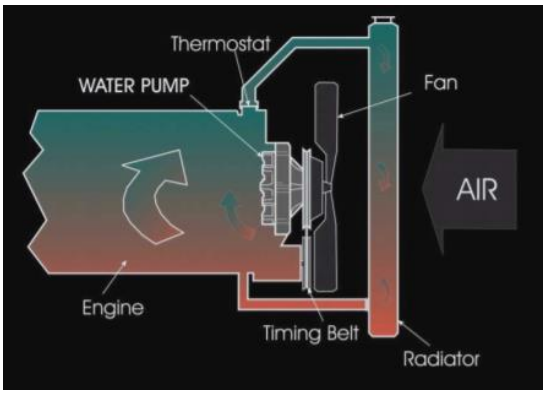
Nombre del Alumno: Alberto Horacio Orozco Ramos

Matrícula: A00831719

20 de Octubre de 2021

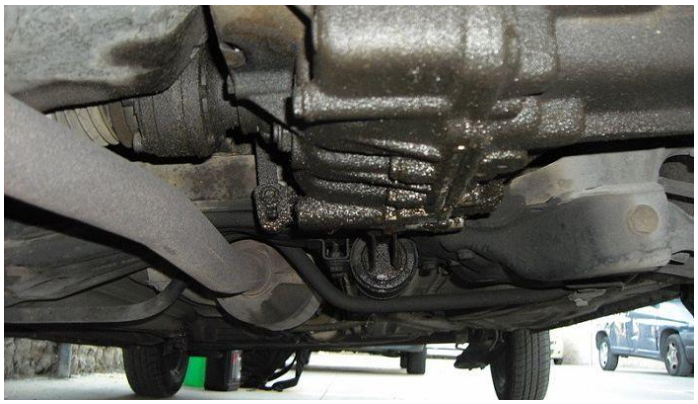
RESOLUCIÓN DEL RETO

Problemática: El sistema de enfriamiento de motor que utiliza agua, como su nombre lo indica, se encarga de regular la temperatura del motor de un vehículo convencional debido a que el proceso de combustión que necesita el vehículo para arrancar y mantenerse encendido requiere de la combustión del combustible que por sí misma genera temperaturas que llegan hasta los 2000°C aproximadamente, esto considerando que se producen gases derivados de la gasolina que concentran aún más temperatura. Ya sea un líquido común como lo es el agua o bien un refrigerante, es el principal factor que reduce la temperatura del motor. Este sale de una bomba de agua que va haciendo un recorrido por todo el sistema y poco a poco logra bajar la temperatura hasta un nivel estable en el que se puedo operar con seguridad. Este sistema se apoya de ventiladores y además existe el termostato, que es un conducto que permite al refrigerante pasar hasta el radiador del motor, ósea gran parte de este. Esto solo ocurre cuando no se ha alcanzado una temperatura que represente un riesgo para el motor y sus componentes, solo cuando es así, el termostato conectará dichos ductos para un enfriamiento más eficiente.

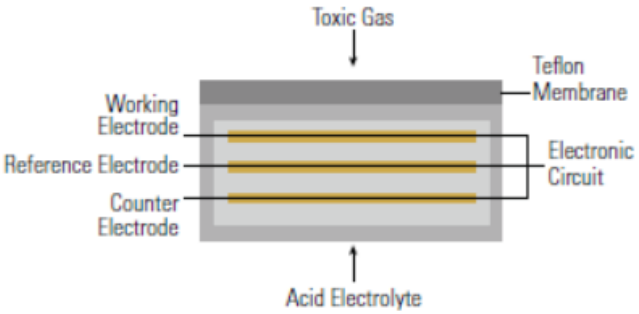


La problemática en sí de este sistema es que, la mayor parte de las personas desconocen muchos de los factores que están en juego con respecto al sistema de enfriamiento del motor, provocando así muchas de las fallas que comúnmente se presentan, acortando así la vida útil del vehículo. Uno de estos tantos factores que juegan un rol muy importante son las juntas de estanqueidad, están son componentes que sirven como una especie de sello que une caras mecanizadas con las cajas de transmisiones, por lo que se prevé una fuga de algún líquido (en este caso aceite), de compresión y/o exceso de presión que pueden provocar que el radiador de refrigeración se hinche, y también que se revienten varias mangueras. Todo esto puede ser causado por un sobrecalentamiento, el motor esta recalentado y muy forzado o bien el motor esta fuera de tiempo, lo cual puede llevar

a problemas de pre-ignición y detonación. Las consecuencias son numerosas, como cualquier componente de un vehículo dañado puede provocar, en este caso existe la posibilidad de que el motor tienda a sobrecalentarse más, pérdida de potencia, presión elevada en el sistema de enfriamiento, fugas de cualquier tipo, el refrigerante llega a cambiar de color, humo gris en el escape y consumos excesivos tanto de agua como de aceite.



Solución: Pensando detenidamente de qué forma podría solucionar o prevenir este tipo de incidentes mediante sensores se me ocurrieron algunas ideas, ya que los mismos autos ya cuentan con medidores que te pueden apoyar a identificar este problema, sin embargo, esto ocurre cuando ya están dañadas las juntas. En base a una pequeña investigación que realicé, puedo determinar de forma teórica que puede ser posible adaptar un sensor electroquímico que lleve el control de oxidación en varias partes del motor. Esto debido a que cuando las juntas se queman gracias al sobrecalentamiento se empiezan a fugar diversos tipos de fluidos, la mayor parte de ellos provocando que componentes como ductos, radiadores, contenedores y hasta sustancias se empiecen a corroer y/o descomponer, entre los defectos que provoca es en gran parte el óxido. Si se llegara a implementar este tipo de sensor, ya sea para detectar en los ductos cierta cantidad de óxido, en el radiador o algún tipo de placas muy delgadas que se ajusten a las mismas juntas, se podría llegar a detectar alguna anormalidad o fallo justo cuando este apenas empiece a afectar el sistema de enfriamiento y se puedan tomar cartas en el asunto cuanto antes. La unidad de medición para detectar el nivel de óxido en los materiales sería por PPM (partes por millón), esta se debe a que los sensores electroquímicos emiten una señal eléctrica intensificada en base a la detección de un gas que pasa por una membrana, parte del sensor, lo que ocasiona un flujo de electrones (un contraelectrodo dirigido hacia un llamado “electrodo de trabajo”), provocando así una corriente eléctrica. Dependiendo de qué tan intensa sea la señal de la corriente se puede determinar el nivel de óxido en el material.



Considerando que requerimos de un dispositivo/procesador/microchip que cumpla con características similares que cumplan los requerimientos del sensor electroquímico, podemos tomar en cuenta el microchip PIC18F4550, que realiza las tareas suficientes para llevar a cabo el trabajo del sensor correctamente. Además,

cabe resaltar que el PIC18F4550 maneja la misma longitud de palabra que el lenguaje ensamblador de MARIE, por lo que no sería tan difícil llevar a cabo este proceso. En la siguiente tabla podemos apreciar las características del PIC18F4550 en comparación a una variante de un ECU con las mismas características base, pero difiriendo con algunas capacidades:

Microchip PIC18F4550	ECU-4784-C25SBE
La capacidad máxima de espacio de memoria que posee es de 16 bits de amplio, mientras que los datos de la memoria RAM son de 8 bits	La capacidad de memoria de este ECU en específico es de aproximadamente 8GB
La capacidad máxima de instrucciones que puede realizar es de 16 bits	Maneja un juego de instrucciones de al menos 16 bits
La máxima frecuencia de trabajo es de 48 MHz	La máxima frecuencia de trabajo es de 1.6GHz
El consumo de energía promedio es de aproximadamente 10-40ma con un voltaje de operación de 4.2V a 5.5V	El consumo de energía promedio de este dispositivo es de 25W

Ahora, para aterrizar un poco estos conceptos y poner en práctica parte de lo que sería esta resolución, haré uso del lenguaje ensamblador MARIE que cumple con características bastante aproximadas a las capacidades que posee el PIC18F4550. En los próximos apartados podrás encontrar tanto el código como evidencia de que realmente funciona para almacenar 50 datos mediante direccionamiento indirecto y le otorga un espacio de memoria específico para cada uno de ellos:

CÓDIGO DE MARIE:

```
ORG 100 // Empezamos en el espacio de almacenamiento 100


FOR, LOAD I // Empezamos el ciclo FOR cargando la variable I a AC

SUBT FIFTY // Realizamos una operación para que se quede guardada en AC y podamos evaluar if (I < 50) con SKIPCOND 800

SKIPCOND 000 // Evaluamos if (I < 50)

JUMP FINISH // De no cumplirse la condición, saltamos a la línea donde se encuentra FINISH

INPUT / El usuario teclea un dato de medición

STOREI LECTOR // Guardamos el dato de medición en memoria mediante Direccionamiento Indirecto

LOAD LECTOR // Cargamos LECTOR a AC

ADD ONEH // Añadimos ONEH(1 base 16) para recorrer el espacio de memoria

STORE LECTOR // Guardamos mediante Direccionamiento Indirecto en el espacio de memoria consecutivo el dato
```

LOAD I // Cargamos I a AC

ADD ONE // Añadimos ONE(1) al valor dentro de AC

STORE I // Guardamos el resultado de la operación anterior en I

JUMP FOR // Saltamos directamente al inicio del ciclo

FINISH, HALT / Terminamos el programa

ONE, DEC 1 // Declaración de la variable ONE = 1 en decimal

ONEH, HEX 1 //Declaración de la variable ONEH = 1 en hexadecimal

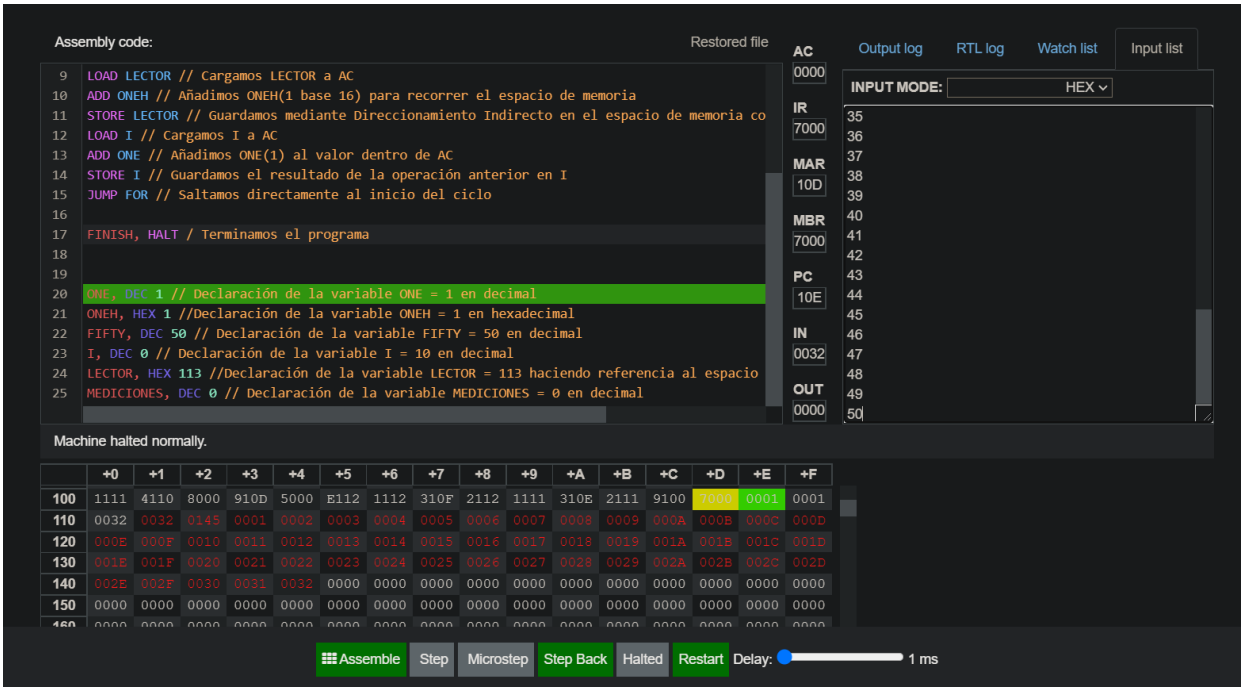
FIFTY, DEC 50 // Declaración de la variable FIFTY = 50 en decimal

I, DEC 0 // Declaración de la variable I = 10 en decimal

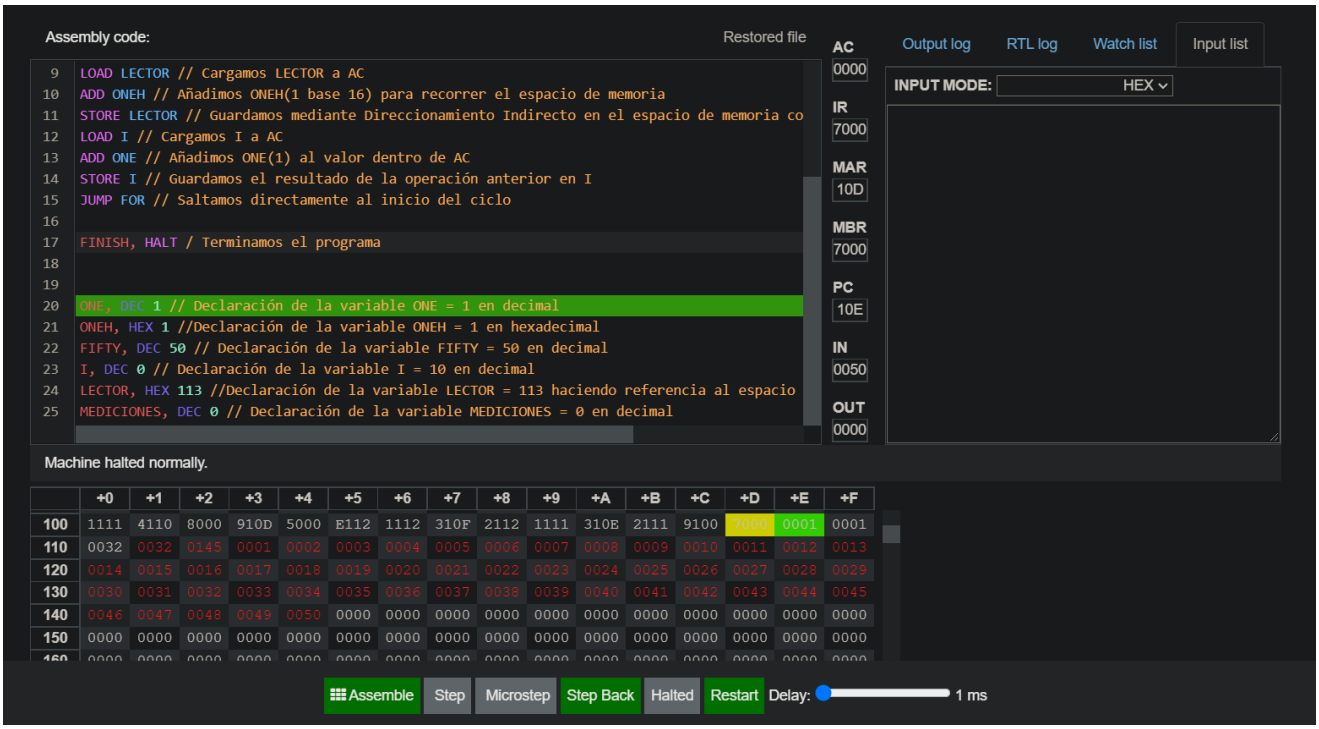
LECTOR, HEX 113 //Declaración de la variable LECTOR = 113 haciendo referencia al espacio de memoria de MEDICIONES

MEDICIONES, DEC 0 // Declaración de la variable MEDICIONES = 0 en decimal

Evidencia de una prueba del código en la consola cargando números del 1 al 50:



Evidencia de captura de pantalla con los valores cargados en la memoria:



Para determinar el tamaño de mi programa para leer el sensor tuve que realizar una pequeña operación de este modo: (líneas de código* 16) + (casillas de memoria *16). En mi caso y tomando en cuenta solamente las variables que estaban siendo guardadas dentro de la memoria (en una casilla), me quedó de la siguiente forma: (19 * 16) + (69 * 16) = 1408 bits = 176 bytes.

Para finalizar con el programa, he decidido añadir algunas especificaciones para que pueda calcular el promedio de todos los 50 datos ingresados, e inclusive hice que el programa no se detenga en ningún momento de solicitar datos y repetir tanto el proceso de almacenamiento (elaborado anteriormente) como el del cálculo del promedio, ósea que el programa nunca va a terminar. Esto tiene una explicación lógica, y es que sí el sensor con el microchip PIC18F4550 integrado va a estar conectado directamente a la consola del automóvil, es obvio que se necesita recabar información iterativamente de forma indefinida a menos de que el suministro de energía deje de proporcionar corriente al sensor electroquímico, y esto solo sucede cuando el auto se encuentra apagado.

Me gustaría mencionar que me hubiera encantado poder implementar un registro por determinado tiempo sobre la cantidad de óxido registrada por el sensor guardada en la misma memoria del PIC18F4550, sin embargo, me encontré con algunas limitantes. En primera instancia, la memoria no creo que baste para almacenar tantos datos, en un punto tendría que empezar a sobrescribir los datos y

probablemente causaría muchos errores dentro del programa tarde o temprano, por lo que se requeriría de una mejor memoria o espacio de almacenamiento. El segundo punto, aunque tuviera la memoria suficiente, me tardaría bastante tiempo para poder implementar esta característica dentro del código realizado en MARIE y debido a la limitante de tiempo que tengo no lo veo factible. Ahora que ya he probado el código con valores numéricos subsecuentes y sé que funciona a la perfección, este no debe tener ningún inconveniente al momento de almacenar valores relacionados a la cantidad de oxidación mediante gas de las juntas de estanqueidad dentro del sistema de enfriamiento del motor, teniendo como unidad las partes por millón (PPM). Consideré en base a una breve investigación que el rango que supone niveles irregulares de oxidación es de entre 250 a 350 PPM. A continuación, adjunto la versión final de mi código hecho en ensamblador:

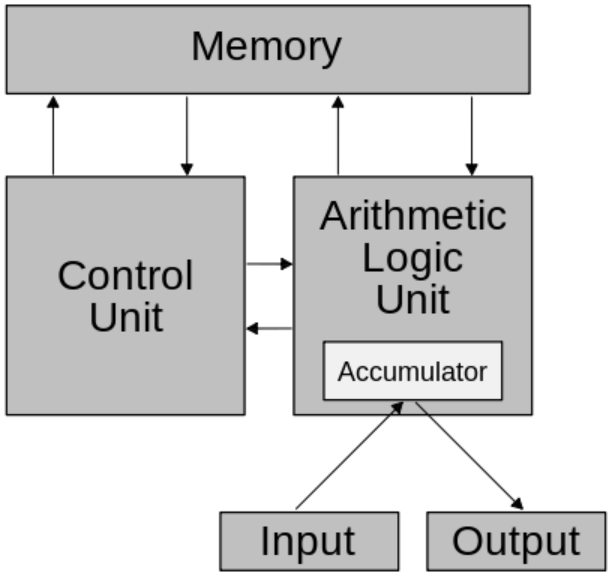
VERSIÓN FINAL DEL CÓDIGO DE MARIE:

```
ORG 100 // Empezamos en el espacio de almacenamiento 100


FOR, LOAD I // Empezamos el ciclo FOR cargando la variable I a AC

SUBT FIFTY // Realizamos una operación para que se quede guardada en AC y podamos evaluar if (I < 50) con SKIPCOND 800

SKIPCOND 000 // Evaluamos if (I < 50)

JUMP FINISH // De no cumplirse la condición, saltamos a la línea donde se encuentra FINISH

INPUT / El usuario teclea un dato de medición

STOREI LECTOR // Guardamos el dato de medición en memoria mediante Direccionamiento Indirecto

LOADI LECTOR // Cargamos el espacio de memoria al que apunta LECTOR a AC

ADD ACUM // Añadimos a AC el valor de ACUM

STORE ACUM // Guardamos el resultado en ACUM

LOAD LECTOR // Cargamos LECTOR a AC

ADD ONEH // Añadimos ONEH(1 base 16) para recorrer el espacio de memoria

STORE LECTOR // Guardamos mediante Direccionamiento Indirecto en el espacio de memoria consecutivo el dato

LOAD I // Cargamos I a AC

ADD ONE // Añadimos ONE(1) al valor dentro de AC

STORE I // Guardamos el resultado de la operación anterior en I

JUMP FOR // Saltamos directamente al inicio del ciclo


WHILE, LOAD ACUM //Cargamos ACUM a AC

SUBT FIFTY //Realizamos una operación para comparar en un if (skipcond)

STORE ACUM //Guardamos valor en la variable ACUM
```

```
SKIPCOND 800 //if (ACUM > FIFTY)
JUMP FINISH //Saltamos a FINISH
LOAD PROM //Cargamos PROM a AC
ADD ONE //Añadimos 1 al valor en AC
STORE PROM //Guardamos valor en la variable PROM
JUMP WHILE //Repetimos el ciclo mediante un salto de línea al principio del mismo

INCREMENTO, LOAD PROM //Cargamos PROM a AC
ADD ONE //Añadimos 1 al valor en AC
STORE PROM //Guardamos el valor en PROM
LOAD ACUM //Cargamos ACUM en AC
SUBT FIFTY //Restamos el valor de AC con FIFTY
STORE ACUM //Guardamos valor en variable ACUM

FINISH, LOAD ACUM //Cargamos ACUM en AC
SKIPCOND 000 //if (X < 0)
JUMP INCREMENTO //Si la condición es falsa, hacemos salto al inicio del ciclo
LOAD PROM //Cargamos PROM en AC
OUTPUT /Desplegamos en la consola el valor en AC
LOAD 0 // Cargamos 0 a AC
STORE I // Reiniciamos el valor de I a 0
STORE ACUM // Reiniciamos el valor de ACUM a 0
STORE PROM // Reiniciamos el valor de PROM a 0
JUMP FOR // Volvemos a repetir todo el proceso con 50 datos nuevas saltando al FOR
//HALT /Finalizamos el programa

ONE, DEC 1 // Declaración de la variable ONE = 1 en decimal
ONEH, HEX 1 //Declaración de la variable ONEH = 1 en hexadecimal
FIFTY, DEC 50 // Declaración de la variable FIFTY = 50 en decimal
I, DEC 0 // Declaración de la variable I = 10 en decimal
ACUM, DEC 0 // Declaración de la variable ACUM = 0 en decimal
PROM, DEC 0 // Declaración de la variable PROM = 0 en decimal
LECTOR, HEX 140 // Declaración de la variable LECTOR = 113 haciendo referencia a un espacio de memoria específico
```

Todo lo anteriormente mencionado en esta pequeña investigación son conceptos muy básicos de todo lo que pude recabar para darle sentido y sustento a una posible hipótesis que puede ser factible para prevenir un desperfecto en cualquier vehículo. Considero que la misma no es perfecta, pero hice todo lo que pude por darle el sustento necesario para que tenga coherencia y sentido, ya que no es fácil hablar de componentes de vehículos y mucho menos de las partes mecánicas del motor debido a que son muy complejas. Añadiendo un punto importante, creo que en general los sensores son una herramienta sumamente fundamental hoy en día, ya que los podemos encontrar en prácticamente todos lados. Literalmente cualquier cosa que haya sido programada para interpretar cierta cosa, patrón o bien comportamiento de cualquier tipo al lenguaje de una computadora y poder responder a estas señales es un sensor. Me gusta mucho como podemos aprovechar los sensores para conectar las computadoras con el mundo exterior, a través de hardware y así sean capaces de realizar cierto trabajo y/o actividad de forma automática, eficaz y sin requerir ninguna intervención humana, claro dependiendo del contexto en que se utilicen.

Me gustaría añadir algunos datos adicionales sobre el modelo de Von Neumann y sus diferencias con los computadores actuales. Básicamente el modelo de Von Neumann se caracteriza por tener 3 partes esenciales: la unidad de CPU (en ella está contenida la unidad de aritmética lógica y la unidad de control que se encarga de guardar registros y contadores de programa), la memoria y los sistemas de entrada y salida.



Las diferencias más destacables entre el modelo mencionado y el actual es que el de Von Neumann utiliza solo un bloque de memoria para datos e instrucciones, mientras que el moderno posee uno para cada uno. El de Von Neumann tiene mejor velocidad que los computadores actuales, esto se debe a que tanto las instrucciones como los datos viajan por el mismo “canal” y no pueden hacerlo al mismo tiempo. El tamaño de las instrucciones se ve limitado por el bus de datos, esto se debe a que el bus y las mismas instrucciones deben tener el mismo tamaño/longitud, mientras que en los sistemas actuales no se necesita acceder a memoria varias veces para buscar instrucciones más complejas. Una ventaja que podemos darle al modelo de Von Neumann es que como los componentes eran mucho menos complejos que ahora, la arquitectura es más simple que las actuales.

Ahora hablando sobre los procesadores mencionados anteriormente, en lo personal, creo que existen diferencias bastante obvias entre los procesadores utilizados en la industria automotriz con el utilizado en la situación problema, en este caso es el PIC18F4550. En el caso de los ECU están precisamente hechos para manejar ciertas características que vayan con las necesidades que deben cubrir en un auto, que son varias. En cambio, el procesador con el que estamos trabajando en el reto cumple con las características básicas, contiene muchas instrucciones, pero no se compara con las que maneja un procesador para un automóvil. De hecho, el procesador para el reto es solo un microchip, el del automóvil es un componente demasiado grande, que consume 5 veces más energía que el microchip, posee más frecuencia de trabajo y por ende debe tener una capacidad de almacenamiento mayor a la del chip, ya que, si hablamos de manipular más componentes, hablamos de más manipulación y registro de datos.

Englobando todo lo dicho anteriormente, ambos poseen los mismos conceptos, las mismas bases y cumplen con las mismas funcionalidades, solo que uno este hecho para operaciones menos exigentes y más simples en comparación con el otro que realiza operaciones a mayor escala y debido a que manipula más componentes (por ende, más variables), requiere potenciar todas sus características al nivel requerido. Por poner un ejemplo relacionado a la propuesta dada en la primera parte de este entregable, el microprocesador integrado en el sensor electroquímico va a manejar ciertas variables, y por supuesto que almacenará una cantidad considerable de información para evaluar el estado de oxidación en áreas específicas del motor de un auto, pero no se compara con todas las tareas que debe atender un procesador ligado a múltiples sensores y aparatos que requieren de atención siempre que el auto se encuentre encendido, y entre más moderno sea el auto, existirán más componentes de los cuales se hará cargo el procesador por casi o enteramente por sí solo, sin intervención humana. Esto haciendo hincapié en que la tecnología existe y ha existido para facilitar y reducir el esfuerzo en tareas que resultan ser tediosas o muy tardadas por si solas.

En conclusión, me gustaría destacar que la tecnología digital ha sido de mucha utilidad con el pasar de los años. En el caso de la industria automotriz, ha hecho posible reducir tareas sumamente laboriosas y complejas hechas por la misma maquinaria, e inclusive rebasar sus límites y hacer que un algoritmo se ocupe del procedimiento pesado. Las empresas que lideran el mercado de los autos deben lanzar productos de calidad, de forma más rápida con requisitos cada que exigen demasiada complejidad, cumplir con normas estrictas y gestionar un crecimiento exponencial a su vez que se cuida de no aumentar el uso de recursos. El impacto que ha traído la tecnología digital a la industria ha hecho posible superar estos retos muy demandantes. Los medios digitales hacen posible transformar y/o implementar conceptos que facilitan mucho al usuario mantener un control más preciso sobre la máquina, e inclusive un algoritmo puede encargarse de múltiples tareas al mismo tiempo y hacerlo de forma exacta y con un margen de error mínimo, en comparación

a realizarlo de forma manual o con componentes físicos que podrían llegar a averiarse o desgastarse con el uso y tiempo. Y en lo personal, esto se refleja muy bien en todos los entregables de este trabajo. Se busca algún tipo de dispositivo que ayude a simplificar, agilizar y/o administrar una tarea que puede llegar a ser muy tediosa y laboriosa por si sola, por lo que a lo largo de la elaboración de este reto tuve que pensar en una problemática que pudiera solucionar con un sensor, para después considerar de la mejor forma cómo actuaría en el entorno que se encontrara y a su vez cómo funcionaria su memoria interna al momento de recabar información y enviarla a la consola del automóvil, todo mediante un proceso digital demasiado complejo si no se aterrizan los conceptos de manera adecuada, pero en verdad pienso que es muy útil aprender estos conceptos, ya que hoy en día vivimos en un mundo digital, dónde no importa en dónde estemos, la tecnología existe en prácticamente cualquier lugar y nos ayuda a hacer nuestra vida mucho más fácil de lo que sería sin estos micro procesos, chips, sensores y demás que realizan cosas muy complicadas por nosotros.

Bibliografía:

- Alvarado, D. (2017). Fallos en el sistema de refrigeración. septiembre 24, 2021, de Fallas en Sistema de Enfriamiento de los Vehículos Sitio web: https://www.nitro.pe/mecanico-nitro/fallos-en-el-sistema-de-refrigeracion.html
- Anónimo. (2011). Óxido de hierro. octubre 16, 2021, de Oxidación en Metales Sitio web: https://www.ecured.cu/%C3%93xido_de_hierro
- Barajas, G. (2021). LA TRANSFORMACIÓN DIGITAL EN LAS EMPRESAS AUTOMOTRICES. octubre 16, 2021, de Digitalización de la Industria Automotriz Sitio web: https://www.metalmecanica.com/temas/La-transformacion-digital-en-las-empresas-automotrices+137401
- Euro Taller. (2015). Averías de la junta de la culata: caras de reparar, peligrosas para el motor. Síntomas de que algo marcha mal. septiembre 24, 2021, de Fallas de Juntas de Motor Sitio web: https://www.eurotaller.com/noticia/averias-de-la-junta-de-la-culata-caras-de-reparar-peligrosas-para-el-motor-sintomas-de-que
- Greelane. (2019). ¿Qué es un agente oxidante? octubre 16, 2021, de Oxígeno como Agente Oxidante Sitio web: https://www.greelane.com/es/ciencia-tecnolog%C3%ADa-matem%C3%A1ticas/ciencia/definition-of-oxidizing-agent-605459/
- Industrial Scientific. (2018). SENSORES ELECTROQUÍMICOS. septiembre 24, 2021, de Sensores Electroquímicos para Detección de Óxido Sitio web: https://www.indsci.com/es/capacitacion/educacion-general-sobre-gases/electrochemical-sensors/

- Runsa Autopartes. (2020). Fallas comunes en el SISTEMA DE ENFRIAMIENTO automotriz (PARTE 2). septiembre 24. 2021, de Fallas en Sistema de Enfriamiento de los Vehículos Sitio web: https://www.youtube.com/watch?v=u9IEJXLKVT8
- Veloso, C. (2016). DIFERENCIAS ENTRE LA MAQUINA DE VON NEUMANN Y LOS COMPUTADORES ACTUALES. septiembre 26, 2021, de Von Neumann y Sistemas Actuales Sitio web: https://www.electrontools.com/Home/WP/diferencias-entre-la-maquina-de-von-neumann-y-los-computadores-actuales/
- Recuperado de: https://ww1.microchip.com/downloads/en/devicedoc/39632c.pdf
- Recuperado de: https://advdownload.advantech.com/productfile/PIS/ECU-4784/file/ECU-4784_DS(060420)-MP20200629144845.pdf