

# **INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE**

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación el 29 de noviembre de 1976.

---

Departamento de Electrónica, Sistemas e Informática

**MAESTRÍA EN DISEÑO ELECTRÓNICO**



## **REPORTE DE FORMACIÓN COMPLEMENTARIA EN ÁREA DE CONCENTRACIÓN EN SISTEMAS EMBEBIDOS Y TELECOMUNICACIONES**

Trabajo recepcional que para obtener el grado de

**MAESTRO EN DISEÑO ELECTRÓNICO**

Presenta: Héctor Adrián Aguirre Díaz

Asesor: Dr. José Luis Chávez Hurtado

San Pedro Tlaquepaque, Jalisco. 3 de Marzo del 2018.



# Contenido

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE.....	I
1. RESUMEN DE LOS PROYECTOS REALIZADOS .....	3
1.1. DISEÑO DE UN REGULADOR DE VOLTAJE TIPO BUCK.....	3
1.1.1 INTRODUCCIÓN.....	3
1.1.2 ANTECEDENTES.....	3
1.1.3 SOLUCIÓN DESARROLLADA .....	4
1.1.4 ANÁLISIS DE RESULTADOS.....	5
1.1.5 CONCLUSIONES .....	5
1.2. MÓDULO DE CONTROL DE CARROCERÍA AUTOMOTRIZ.....	6
1.2.1 INTRODUCCIÓN.....	6
1.2.2 ANTECEDENTES.....	6
1.2.3 SOLUCIÓN DESARROLLADA .....	6
1.2.4 ANÁLISIS DE RESULTADOS.....	7
1.2.5 CONCLUSIONES .....	8
1.3. CODIFICADOR DE CANAL EN UN SISTEMA DE COMUNICACIÓN DIGITAL .....	8
1.3.1 INTRODUCCIÓN.....	8
1.3.2 ANTECEDENTES.....	8
1.3.3 SOLUCIÓN DESARROLLADA .....	9
1.3.4 ANÁLISIS DE RESULTADOS.....	10
1.3.5 CONCLUSIONES .....	10
2. CONCLUSIONES.....	10
APÉNDICES .....	11
A. DISEÑO DE UN REGULADOR DE VOLTAJE TIPO BUCK .....	13
B. MÓDULO DE CONTROL DE CARROCERÍA AUTOMOTRIZ.....	27
C. CODIFICADOR DE CANAL EN UN SISTEMA DE COMUNICACIÓN DIGITAL .....	57



# Introducción

Al estudiar la Maestría en Diseño Electrónico decidí enfocarme en el área de concentración de sistemas embebidos y telecomunicaciones ya que personalmente tenía el interés de los temas que se trataban y a su vez se apegaban al ámbito laboral que actualmente desempeño. Los proyectos dentro de esta área de concentración que se incluyen en este reporte son:

Materia: Diseño de sistemas analógicos basados en dispositivos comerciales.

Proyecto: Diseño de un regulador de voltaje tipo buck.

Materia: Ingeniería de software en ambientes embebidos.

Proyecto: Módulo de control de carrocería automotriz.

Materia: Introducción a las telecomunicaciones digitales y analógicas.

Proyecto: Codificador de canal en un sistema de comunicación digital.

Elegí estos tres proyectos porque representaron un mayor impacto en mi formación tanto académica como laboral. El primero de ellos es el más apegado a mi área laboral ya que formo parte de un equipo de validación eléctrica de reguladores de voltaje, tanto en estabilidad como en consumo e integridad de potencia. Este proyecto me ayudó a reafirmar los conceptos, a resolver dudas y también por otra parte a generar ideas para la mejora de estos circuitos. El segundo proyecto, personalmente fue el de mayor satisfacción porque está muy apegado al área automotriz, por la cual siempre he tenido interés. La idea de ingeniería de software que yo tenía era diferente, pero al realizar todo el proceso de implementación del *Body Control Module*, pude direccionar los conceptos e ideas hacia un mismo punto. El tercer proyecto fue totalmente de aprendizaje ya que sólo sabía conceptos muy básicos. El uso de la herramienta de Simulink, fue lo que más me impactó por el poder de procesamiento y modelado de circuitos digitales que tiene.

En general estos proyectos aportaron conocimientos claves para la oportunidad de crecimiento laboral donde actualmente desempeño mis funciones como ingeniero de validación de potencia para microprocesadores de tecnología de punta.



# 1. Resumen de los proyectos realizados

Esta sección describe de manera general los proyectos mencionados anteriormente.

## 1.1. Diseño de un regulador de voltaje tipo buck

### 1.1.1 Introducción.

Existen varias topologías de diseño para reguladores de voltaje DC-DC, donde la intención es controlar y entregar un voltaje a la salida, menor al voltaje de entrada. Dependiendo de los requerimientos de potencia es donde cada topología tiene sus ventajas y desventajas. Por ejemplo para un sistema de baja potencia, una forma simple de regular y reducir el voltaje de salida es utilizar un divisor de tensión, pero cuando el requerimiento de potencia es mayor, la pérdida de energía de este regulador lineal se concentra en forma de calor. Es por esto que para sistemas de corriente directa de alta potencia, es mejor utilizar la topología tipo *buck*, ya que por conmutación a ciertos ciclos de trabajo es posible regular el voltaje a la salida a diferentes cargas, permitiendo esto una mejor eficiencia.

### 1.1.2 Antecedentes.

Para la implementación de este proyecto se utilizó la topología tipo *buck*, ya que se pretendía utilizar este regulador de voltaje para dispositivos en una computadora, donde la entrada de voltaje usualmente es de 12V y los dispositivos como microprocesadores, interfaces USB, DDR, etcétera, requieren de voltajes menores. Los microprocesadores pueden llegar a consumir alta potencia, aquí es donde este diseño será capaz de mantener un voltaje continuo a altas corrientes debido al sistema de compensación agregado a su retroalimentación, donde podrá reaccionar a un cambio abrupto de voltaje, permitiendo así una entrega de potencia continua. Por otro lado se sabe que los reguladores de voltaje tipo *buck* alcanzan una eficiencia comúnmente mayor a 90%, permitiendo así un mejor desempeño a una menor potencia.

### 1.1.3 Solución desarrollada

El diseño se desarrolló para soportar voltajes de entrada de 9V a 15V, y mantener un voltaje continuo a la salida de 1.5V, con una variación de rizado no máxima del 10%. Con base a esto se realizó un diseño que se conformó de varias etapas: modulación, filtro de salida, etapa de potencia y compensación.

Para la etapa de modulación, se diseñó un PWM (*Pulse Width Modulator*) donde teniendo como entradas una señal triangular y una señal de error de voltaje que viene de la etapa de compensación, a un comparador, este pueda controlar el ciclo de trabajo de su señal cuadrada a la salida, que conmutará los transistores de la etapa de potencia, conformada por dos transistores NMOS conectados al voltaje de entrada, que al final definirán el nivel de voltaje de salida de este regulador. Para el filtro de salida se optó por utilizar un filtro LRC, el cual consiste en un inductor y un capacitor conectados en serie con una resistencia, teniendo como entrada el nodo de fase de la etapa de potencia, mientras que la salida es el voltaje que se proveerá al dispositivo.

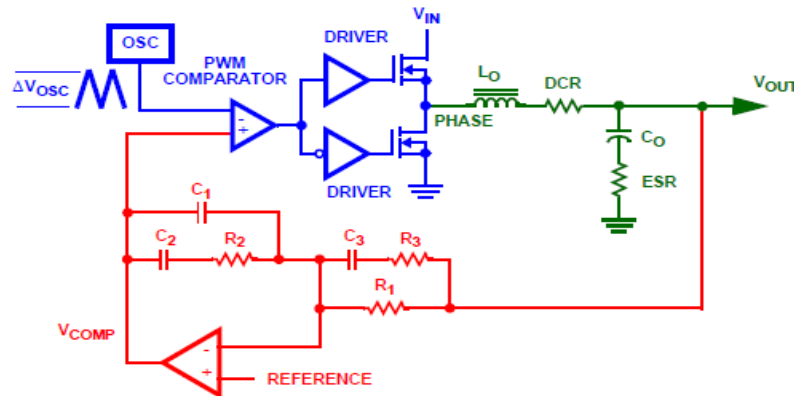


Figura 1. Regulador de voltaje implementado.

Por último para la etapa de compensación se utilizó un compensador tipo 3, que permite que el sistema responda a cambios abruptos de voltaje debido a altas demandas de corriente en la salida, dando un *boost* de 180 grados a la fase utilizando dos ceros. Dicho *boost* es necesario para contrarrestar efectos de resonancia poco amortiguada en el filtro de salida. Este compensador tiene como entrada un voltaje de referencia y a su vez es retroalimentado por el censado de voltaje que hay a la salida. En la figura 1 se puede observar el diseño completo del regulador de voltaje.



### 1.1.4 Análisis de resultados

Utilizando la función de transferencia del compensador, los parámetros de voltaje y corriente tanto de entrada como de salida, se pudo obtener la ecuación de ganancia que nos ayudó a encontrar los valores de resistencias y capacitores del compensador para obtener un sistema con un margen de fase y ancho de banda que permitieran una salida estable. Para esta implementación se tuvo que partir de valores iniciales para los componentes discretos del filtro de salida (inductor y capacitor), en donde después de varias simulaciones se hicieron ajustes necesarios para llegar a los valores finales. La implementación fue puesta a prueba usando el software de LT Spice, obteniendo los siguientes resultados a la salida del regulador:

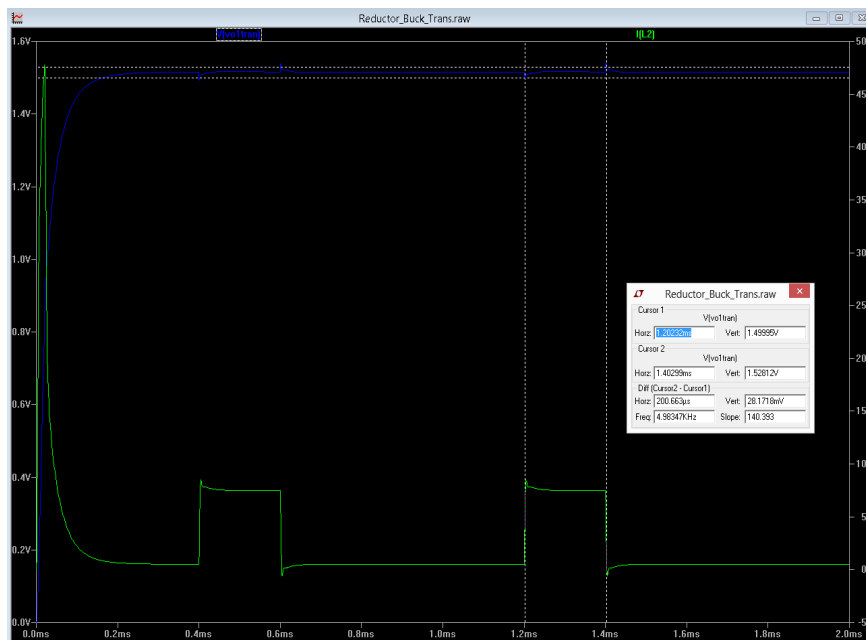


Figura 2. Respuesta transitoria del regulador de voltaje.

### 1.1.5 Conclusiones

Al realizar este proyecto pude concretar mis conceptos de diseño de reguladores de voltaje, y así ponerlos en práctica ya que en mi área laboral sólo se hace la validación. Las simulaciones

que se hicieron en LT Spice fueron la parte clave para encontrar los valores finales de los componentes de este regulador.

## **1.2. Módulo de control de carrocería automotriz**

### **1.2.1 Introducción**

En este proyecto se llevó a cabo el desarrollo de un módulo de control de carrocería automotriz, BCM (Body Control Module) por sus siglas en inglés, el cual se encarga de controlar y monitorear la mayoría de dispositivos electrónicos en un vehículo.

### **1.2.2 Antecedentes**

El BCM, es el módulo que interactúa con el entorno del auto de manera directa, esto lo hace obteniendo información de señales digitales y analógicas provenientes de sensores encargados del monitoreo y control de ciertas actividades del auto. Es por esto que se requieren puertos de entrada y de salida, digitales y analógicos. Los puertos de entrada analógicos sirven para obtener información de velocidad, humedad, carga de batería, temperatura, nivel de gasolina, etcétera y en el caso de los puertos digitales, estos reciben información por ejemplo de la posición de la llave de encendido del auto, el status del pedal del embrague, cinturones de seguridad, etcétera. El BMC en base a esta información debe ser capaz de enviar mensajes al clúster, para mostrarle al conductor el status del auto, o por otro lado para el control del aire acondicionado dependiendo de la temperatura interior del auto, etc. Debido a todos estos requerimientos es necesario utilizar un microcontrolador que pueda ser capaz de procesar todas las señales.

### **1.2.3 Solución desarrollada**

El desarrollo de este proyecto se llevó a cabo en las siguientes etapas: requerimientos técnicos, requerimientos del usuario, modo de fallas y análisis de efectos, diagrama de calidad, arquitectura de software, arquitectura del sistema, generación del banco de pruebas, codificación, implementación y ejecución de pruebas funcionales.

La etapa de requerimientos es una de las principales porque aquí se define el alcance del proyecto, donde tanto el desarrollador como el cliente están de acuerdo. Estos requerimientos deben cumplir ciertos lineamientos, no tienen que ser ambiguos, factibles, etc.

Para la etapa de modo de fallas y análisis de efectos, es necesario tener un diagrama de como acatar cualquier falla en específico que se presente, por ejemplo, si existe una falla con la batería del auto, el efecto es que no podrá ser encendido, este puede ser un problema de fábrica, o que se dañó por algún corto, y para resolver el problema es necesario poner una batería nueva.

El diagrama de calidad ayuda tanto al desarrollador como al cliente, porque aquí se especifica que es lo que se espera del proyecto, porque puede suceder en algunos casos que el cliente espera más funciones que las que se habían estipulado. La mayoría de las funciones son estipuladas en este diagrama.

La arquitectura del sistema contempla tanto el software como el hardware. Para la cuestión del software su utilizó como referencia el modelo AUTOSAR, ya que es un estándar en la industria automotriz. Para la arquitectura de hardware, se pudo observar diferentes opciones, ya que existen muchos dispositivos capaces de realizar la misma función.

La implementación es la etapa que toma más tiempo ya que se tiene que tomar en cuenta todas las cuestiones, es decir el procesamiento de información por todas las capas de software. Por ejemplo la capa de bajo nivel llamada MCU2 que es la encargada de enviar y recibir datos de los dispositivos de hardware externos tiene que procesar los datos mediante sus registros de lectura y escritura para poder ser enviados hasta la capa de aplicación.

#### **1.2.4 Análisis de resultados**

Para analizar los resultados primero se tuvo que haber generado las camas de pruebas, que por lo general se enfocan en los algoritmos del software. Se desarrolló un prototipo simple que permitió ejecutar las camas de prueba, lo cual ayudó a reducir el tiempo de validación. Después de correr las camas de pruebas nos pudimos dar cuenta del funcionamiento de este módulo. El haber utilizado la arquitectura AUTOSAR fue de gran ayuda ya que es un estándar automotriz, el cual ya tiene sus limitaciones bien definidas, y a su vez facilita el trabajo en equipo reutilizando el código y evitando dobles esfuerzos.

### **1.2.5 Conclusiones**

Este proyecto fue de gran satisfacción personal porque abarcó un área que me gustaría dedicar en el algún tiempo ya que como mencioné anteriormente es de mi interés. El haber conocido AUTOSAR que es un estándar de la industria me será de gran ayuda en un futuro para simplificar el trabajo y esfuerzo que toma hacer un proyecto.

## **1.3. Codificador de canal en un sistema de comunicación digital**

### **1.3.1 Introducción**

Las partes fundamentales de un sistema de comunicación son el codificador de canal, el canal de comunicación y el decodificador de canal. Es importante codificar la información ya que se puede disminuir la probabilidad de errores provocados por los efectos del ruido del canal, para esto se agrega información redundante que permite al receptor detectar errores y corregirlos. En este proyecto se llevó a cabo el desarrollo de un codificador de canal para reducir la probabilidad de errores.

### **1.3.2 Antecedentes**

Existe un parámetro en los sistemas de comunicaciones que indica la relación señal a ruido de un canal de comunicación, SNR (Signal to Noise Ratio), este parámetro puede ser afectado por el desvanecimiento en la información enviada. Al tener un nivel inadecuado de SNR no se alcanzarán a corregir los errores en la información. Con el fin de detectar y corregir los bits erróneos, se aplica una serie de transformaciones a la señal transmitida, el cual permitirá reducir la probabilidad de error, o garantizar un SNR necesario para cierta tasa de error, transformando secuencias binarias en secuencias que incluyan una redundancia estructurada, estos bits de redundancia, son utilizados para detectar y corregir errores.

### 1.3.3 Solución desarrollada

Para este proyecto se desarrolló un sistema de comunicación básico en la plataforma de simulación de Matlab. Donde se tiene un generador de código y un canal binario simétrico que permite configurar directamente la probabilidad de error que se desea sobre los datos binarios recibidos. En la figura 3 se muestra la implementación del mismo en simulink.

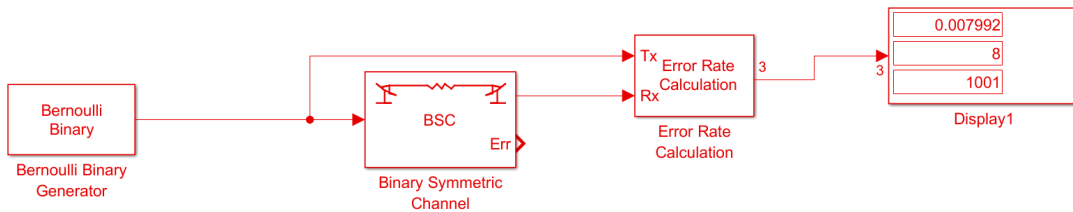


Figura 3. Sistema de comunicación desarrollado en simulink

Se ejecutaron varias simulaciones para diversos valores de probabilidad de error del canal, generando por lo menos un total de 1000 bits de información. En la siguiente tabla se muestra la probabilidad de error obtenida para cada caso.

Probabilidad de Error Configurada	Probabilidad de Error Obtenida
0.005	0.007992
0.01	0.01399
0.015	0.01998
0.025	0.03297
0.05	0.05894
0.1	0.1149
0.15	0.1608
0.2	0.2168
0.25	0.2637
0.3	0.3287

Para reducir la probabilidad de error obtenida, posteriormente se amplió el sistema, añadiendo un codificador y decodificador de canal de Bloques de Hamming, que por omisión su configuración es de un código (7,4). En la figura 4 se puede observar el sistema anterior pero ahora con un codificador y un decodificador.

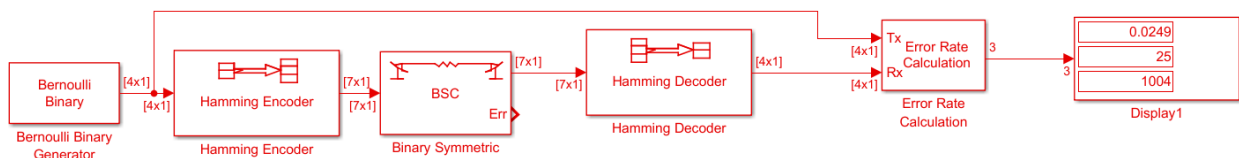


Figura 4. Sistema de comunicación con codificador y decodificador tipo Hamming.

### 1.3.4 Análisis de resultados

Se volvieron a realizar las mismas simulaciones, y ahora se puede observar en la siguiente tabla, que la probabilidad de error obtenida ahora es mucho menor.

Probabilidad de Error Configurada	Probabilidad de Error Obtenida
0.005	0
0.01	0.000996
0.015	0.000996
0.025	0.000996
0.05	0.0249
0.1	0.07271
0.15	0.1355
0.2	0.2141
0.25	0.2739
0.3	0.3307

### 1.3.5 Conclusiones

Al realizar la comparación de los resultados de los sistemas, queda claro que la codificación Hamming es muy útil para reducir la probabilidad de error obtenida, siempre y cuando las probabilidades de error estén dentro de un rango, ya que de tener probabilidades mayores la codificación de Hamming no será tan eficiente. Este proyecto me ayudó a reafirmar los conceptos de comunicaciones, ya que en la universidad solo se cubrió lo básico, y también me ayudó a comprender el porqué de los errores que tenemos actualmente en la validación de diferentes interfaces en donde trabajo.

## 2. Conclusiones

El haber desarrollado los proyectos anteriormente mencionados, me dieron una oportunidad de crecimiento profesional, ya que los conocimientos adquiridos después de la implementación de cada uno, fueron claves para resolver problemas porque ahora entiendo mejor el funcionamiento de ciertos dispositivos que son validados en la empresa donde trabajo y la vez me han ayudado para tomar decisiones finales ya que las simulaciones algunas veces han sido muy cercanas a los sistemas reales. El uso de herramientas como Matlab, Simulink, son ahora cruciales en la industria, ya que para mantener un puesto dentro de la competencia, se tiene que confiar más en la validación que se hace antes de diseñar o implementar el circuito, o dispositivo.

# Apéndices





## A. DISEÑO DE UN REGULADOR DE VOLTAJE TIPO BUCK

### Contenido

<i>Descripción general</i> .....	13
<i>Modulador</i> .....	13
<i>Filtro de salida</i> .....	14
<i>Red de compensación</i> .....	15
<i>Sistema completo</i> .....	17
<i>Especificaciones del regulador</i> .....	18
<i>Inductor</i> .....	18
<i>Capacitor</i> .....	18
<i>Controlador</i> .....	19
<i>Compensador</i> .....	20
<i>Simulación</i> .....	22
<i>Costos y materiales</i> .....	25

### *Descripción general.*

Un regulador tipo buck es un convertidor de potencia, produciendo a su salida un voltaje continuo menor que su entrada. En sí, es una fuente conmutada con dos dispositivos semiconductores, siendo ambos MOSFETs, un inductor y un capacitor. El conmutar tiene ventajas sobre un divisor de tensión, tal como el ahorro de energía en forma de calor y eficiencia.

Se contempla una entrada variable de 9V a 15V, teniendo a la salida un voltaje de 1.5V, con una variación máxima del 15%. El regulador debe contar con las siguientes etapas:

- Modulador.
- Filtro de salida.
- Compensación.

### *Modulador.*

La entrada del modulador es la salida del amplificador de error, la cual es usada para comparar la salida con la referencia, mientras que su salida es el nodo de fase. La ganancia del modulador es el voltaje de entrada dividido por el voltaje de pico a pico del oscilador; dicho dato puede obtenerse de la hoja de datos del controlador a usarse.

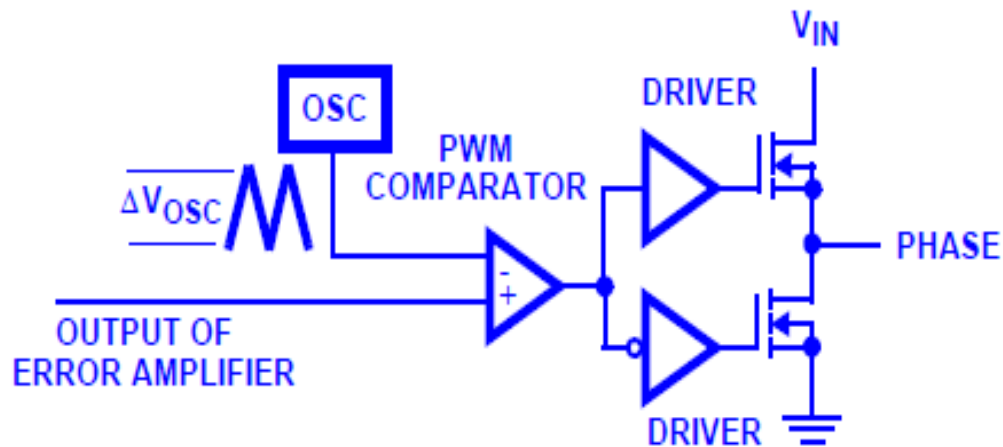


Figura 1: modulador.

La ganancia del modulador está dada por la siguiente ecuación:

$$Gain_{Mod} = \frac{V_{in}}{\Delta V_{osc}}$$

### *Filtro de salida.*

El filtro de salida consiste en el inductor de salida, así como en la capacitancia. Se debe incluir tanto la resistencia DCR en serie con el inductor, así como la ESR del capacitor. La entrada del filtro de salida será el nodo de fase de la etapa del modulador, mientras que la salida de esta etapa es la salida del regulador.

La función de transferencia está dada por la siguiente ecuación:

$$Gain_{Filter} = \frac{1 + s * ESR * C_{out}}{1 + s * (ESR + DCR) * C_{out} + s^2 L_{out} * C_{out}}$$

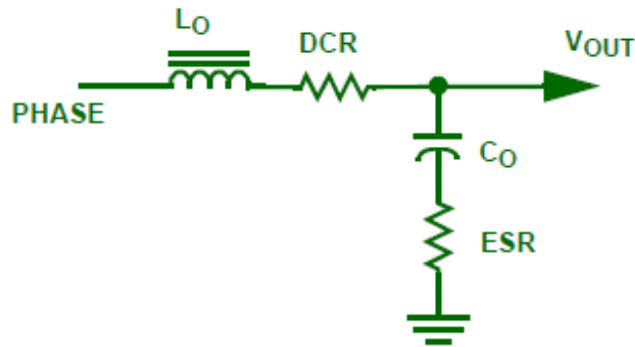


Figura 2: filtro de salida.

### *Red de compensación.*

Cerrar el lazo de control permite al regulador ajustar perturbaciones de la carga o cambios en el voltaje de entrada. Una compensación adecuada permite un ancho de banda predecible y estabilidad incondicional. Utilizando una compensación tipo 3 se le da forma al perfil de la ganancia con respecto a la frecuencia, dando un boost de 180 grados a la fase usando dos ceros. El boost es necesario para contrarrestar efectos de resonancia poco amortiguada ubicada en el filtro de salida en el polo doble.

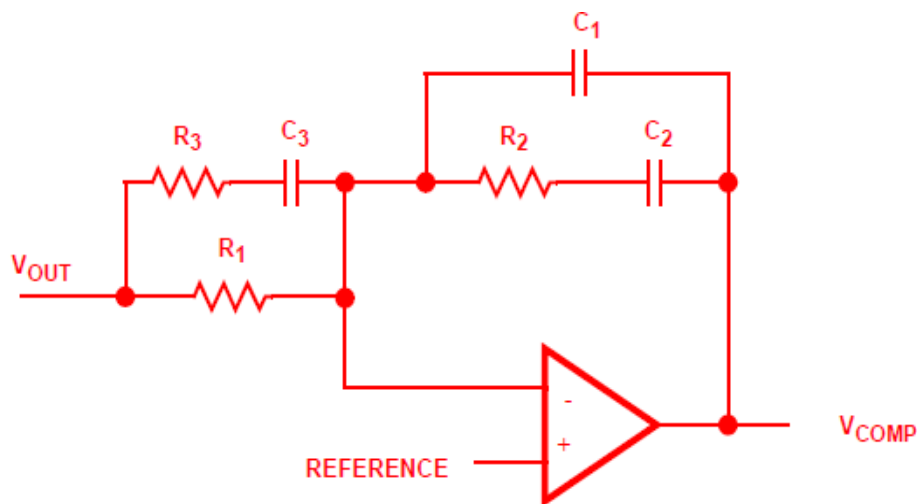


Figura 3: red de compensación.

El procedimiento para cálculo de componentes es el siguiente:

- Escoger un valor para R1 entre 2k y 5KOhms.
- Escoger una ganancia de R2/R1 la cual mueva la ganancia de lazo abierto al ancho de banda deseado, permitiendo que el crossover de 0dB ocurra en el rango de frecuencias donde la compensación de tipo 3 tenga su segunda ganancia plana. R2 está dada por la siguiente ecuación:

$$\bullet \quad R_2 = \left( \frac{DBW}{F_{LC}} \right) * \frac{\Delta V_{OSC}}{V_{IN}} * R_1$$

- Cálculo de C2 poniendo el cero al 50% de la frecuencia del doble polo del filtro de salida.

$$\bullet \quad C_2 = \frac{1}{\pi * R_2 * F_{LC}}$$

- Cálculo de C1 poniendo el primer polo en la frecuencia cero del ESR:

$$C_1 = \frac{C_2}{2\pi * R_2 * C_2 * F_{ESR} - 1}$$

- Poner el segundo polo a la mitad de la frecuencia de switching. Poner también el segundo cero en el polo doble del filtro de salida, obteniendo las siguientes ecuaciones:

$$\bullet \quad R_3 = \frac{R_1}{\frac{F_{SW}}{2 * F_{LC}} - 1}$$

$$\bullet \quad C_3 = \frac{1}{\pi * R_3 * F_{SW}}$$

- La ganancia de la compensación debe ser comparada con la ganancia de lazo abierto del amplificador de error. La ganancia de la compensación no deberá exceder dicha ganancia del amplificador de error dado que ésta es el factor delimitante de la compensación.
- La ganancia de la red de compensación está dada por la siguiente ecuación:

$$\bullet \quad Gain_{comp} = \frac{R_1 + R_3}{R_1 * R_3 * C_1} * \frac{\left( s + \frac{1}{R_2 * C_2} \right) * \left( s + \frac{1}{(R_1 + R_3) * C_3} \right)}{s * \left( s + \frac{C_1 + C_2}{R_2 * C_1 * C_2} \right) * \left( s + \frac{1}{R_3 * C_3} \right)}$$

La ganancia en decibelios, así como la fase se muestra en la siguiente gráfica de Bode:

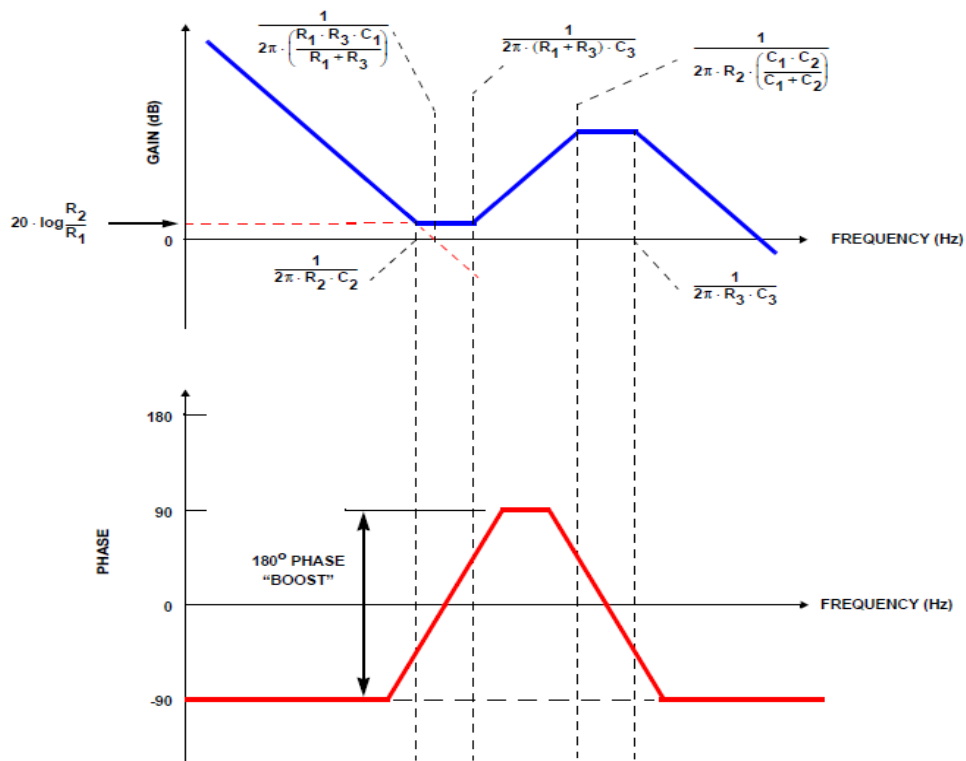


Figura 4: ganancia asintótica de Bode.

*Sistema completo.*

Integrando las tres partes ya mencionadas, se cuenta con el regulador de voltaje completo:

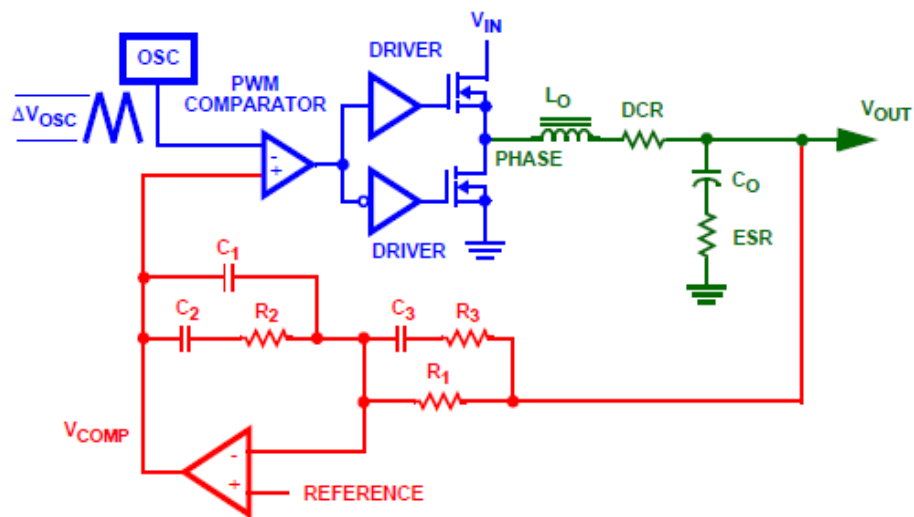


Figura 5: regulador completo.

La ganancia está dada por la siguiente ecuación:

$$Gain_{System} = \frac{R_1 + R_3}{R_1 * R_3 * C_1} * \frac{\left(s + \frac{1}{R_2 * C_2}\right) * \left(s + \frac{1}{(R_1 + R_3) * C_3}\right)}{s * \left(s + \frac{C_1 + C_2}{R_2 * C_1 * C_2}\right) * \left(s + \frac{1}{R_3 * C_3}\right)} * \frac{V_{in}}{\Delta V_{osc}} * \frac{1 + s * ESR * C_{out}}{1 + s * (ESR + DCR) * C_{out} + s^2 * L_{out} * C_{out}}$$

### *Especificaciones del regulador.*

El regulador de voltaje cumplirá con las siguientes especificaciones:

- El rango del voltaje de entrada es de 9V a 15V.
- El voltaje de salida debe de ser menor a 1.5V con una variación máxima de tolerancia, rizado y transitorio de 5%.
- La corriente máxima de salida es de 10A, con un escalón máximo de 7A.
- Frecuencia fija de 500kHz.

### *Inductor.*

Para escoger un valor inicial del inductor, se parte de la siguiente fórmula:

$$\Delta I = \frac{V_{OUT}}{L * f_{SW}} \left(1 - \frac{V_{OUT}}{V_{IN}}\right)$$

Donde  $\Delta I$  es la variación máxima de corriente en el inductor, que para este diseño es de 7A.

$$L = \frac{1.5V}{7A * 500KHz} \left(1 - \frac{1.5}{15}\right) = 0.386\mu H$$

Puesto que la frecuencia es fija y se necesita una reducción en el rizado de corriente en el inductor, se incrementó su valor a 1.3 $\mu H$  con un DCR de 2.3m $\Omega$ .

### *Capacitor.*

El capacitor de salida está determinado por la resistencia en serie equivalente ESR para minimizar el rizado del voltaje de salida durante los cambios de corriente en la carga. La fórmula del voltaje es la siguiente:

$$\Delta V_{OUT} \leq \Delta I_L \left(ESR + \frac{1}{8 * f_{SW} * C_{OUT}}\right)$$

Para  $\Delta V_{OUT}$  de 75mV y despejando para  $C_{OUT}$ :

$$C_{OUT} = \frac{\Delta I_L}{(8 * f_{SW}) * (\Delta V_{OUT} - \Delta I_L * ESR)}$$

Si se considera que la capacitancia equivalente utilizará más de 1 capacitor para reducir el ESR que se estima que es de 3m $\Omega$ :

$$C_{OUT} = \frac{7A}{(8 * 500KHz) * (75mV - 7A * 3m\Omega)} = 33.94\mu F$$

Para tener un rizado de voltaje pequeño durante el transitorio se incrementó el tamaño de la capacitancia de salida por dos capacitores de 470  $\mu F$  con un ESR total de 1.5m $\Omega$ . Para el caso del capacitor de entrada, este debe tener un ESR bajo para minimizar la caída de voltaje cuando los MOSFETs se encienden y su capacitancia debe ser lo suficientemente grande para poder mantener el voltaje de entrada hasta que la fuente pueda responder. Se escoge un capacitor de tipo electrolítico de 220  $\mu F$  con un voltaje por lo menos 1.4 veces más grande que el voltaje máximo de entrada, es decir a 25.

### *Controlador.*

El controlador a utilizar es el modelo LTC3775 de Linear Technology, el cual cuenta con las siguientes características:

- High Frequency Synchronous Step-Down Voltage Mode DC/DC Controller
- Amplio rango de voltaje de entrada, de 4.5V hasta 38V.
- Compensación tipo feed-forward con un amplificador de error de gran ancho de banda para una respuesta rápida en modo transitorio.
- Una salida de voltaje de 0.6 a 0.8 veces el voltaje de entrada.
- Un voltaje de referencia 0.6V.
- Limitador de corriente programable.
- Censado de corriente.
- Soft-start programable.
- La frecuencia de oscilación es de 500KHz.

El controlador tiene un voltaje de oscilación de PWM de 1.25V. La ganancia de la etapa de modulación está dada por la siguiente ecuación:

$$GANANCIA_{MOD} = \frac{V_{in}}{\Delta V_{osc}}$$

Además, utiliza un amplificador operacional interno para censar el voltaje de salida  $V_{OUT}$ . En estado estable la ganancia de salida del regulador es:

$$V_{OUT} = V_{REF} * \left( 1 + \frac{R_A}{R_B} \right)$$

Con  $R_A = 2.05K\Omega$  y  $R_B = 1.37K\Omega$  y el voltaje de referencia interno  $0.6V$ , se tiene que:

$$V_{OUT} = 0.6V * \left( 1 + \frac{2.05K\Omega}{1.37K\Omega} \right) = 1.4978V \approx 1.5V$$

El esquemático del controlador se muestra en la siguiente figura:

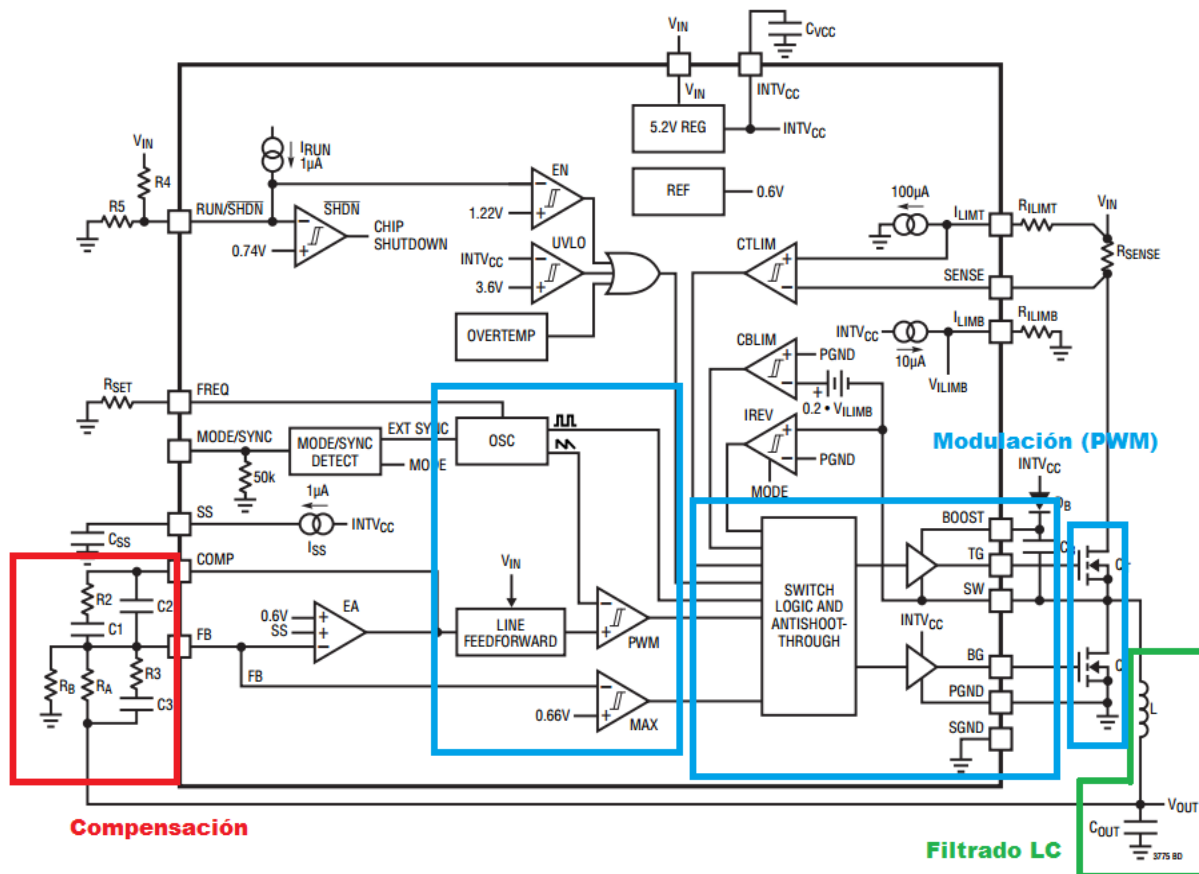


Figura 6: esquemático del controlador.

*Compensador.*



La red de compensación utiliza dos ceros para dar un boost a la fase del sistema de 180°, con el fin de contrarrestar los efectos de la resonancia ocasionados por el doble polo del filtro LC. Un margen de fase de mayor a 45° representa un sistema estable.

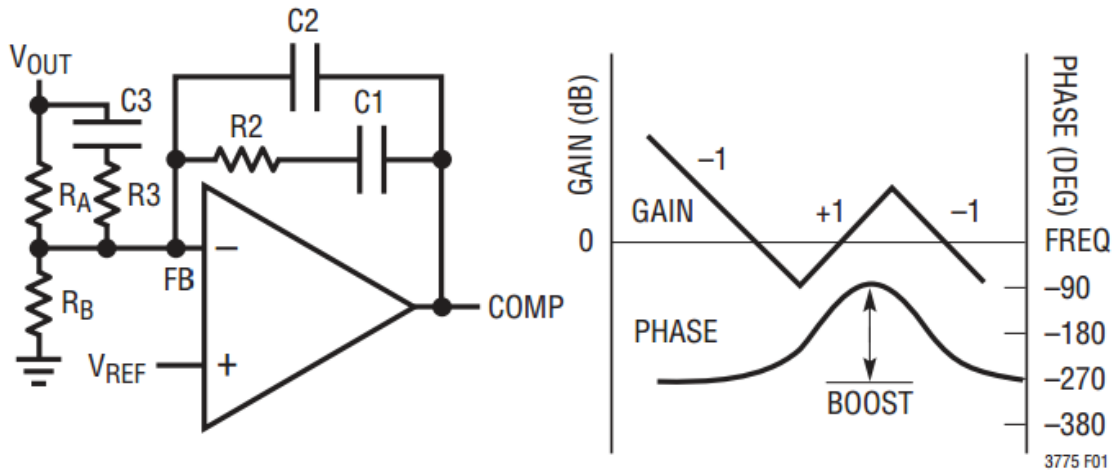


Figura 7: compensador.

Una vez elegida la relación de las resistencias de feedback  $R_A$  y  $R_B$ , basándose en  $R_A$  se puede calcular  $R_2$ . La relación  $\frac{R_2}{R_A}$  recorrerá la ganancia de lazo abierto hasta el ancho de banda deseado. Para un ancho de banda deseado  $DBW=90KHz$ ,  $\Delta V_{OSC}=1.25V$  y  $R_A=2.05K\Omega$ , se tiene que:

$$R_2 = \left( \frac{DBW}{F_{LC}} \right) * \frac{\Delta V_{OSC}}{V_{IN}} * R_1$$

Donde  $F_{LC}$  es la frecuencia de resonancia dada por:

$$F_{LC} = \frac{1}{2\pi\sqrt{LC}} = \frac{1}{2\pi\sqrt{1.3\mu H * 940\mu F}} = 4.53KHz$$

$$R_2 = \left( \frac{90KHz}{4.53KHz} \right) * \frac{1.25V}{15V} * 2.05K\Omega = 3.394 K\Omega \approx 3.4K\Omega$$

Si se coloca el primer cero en  $F_{Z1} = F_{LC}$ , el valor de  $C_1$  está dado por la siguiente relación:

$$C_1 = \frac{1}{2\pi * R_2 * F_{LC}} = \frac{1}{2\pi * 3.4K\Omega * 4.53KHz} = 10.3nF \approx \mathbf{0.01\mu F}$$

Colocando el primer polo en la frecuencia  $f_{ESR}$  de 0.079MHz, se tiene para  $C_2$ :

$$C_2 = \frac{C_1}{2\pi * R_2 * C_1 * F_{ESR} - 1} = \frac{10.3nF}{2\pi * 3.4K\Omega * 10.3nF * 0.079MHz - 1} = 629pF$$

El segundo polo se coloca a la mitad de la frecuencia de conmutación, obteniendo así los siguientes valores para el resto de los componentes discretos:

$$F_{P2} = F_{sw} = \frac{500KHz}{2} = 250KHz.$$

$$R_3 = \frac{R_A}{\frac{F_{sw}}{2 * F_{LC}} - 1} = \frac{2.05K\Omega}{\frac{250KHz}{4.53KHz} - 1} = 37.8\Omega$$

$$C_3 = \frac{1}{\pi R_3 * 250KHz} = \frac{1}{\pi * 37.8\Omega * 250KHz} = 34.nF$$

### Simulación.

La simulación del regulador de voltaje fue realizada mediante el software de LT Spice, cuyo esquemático se muestra en la siguiente figura:

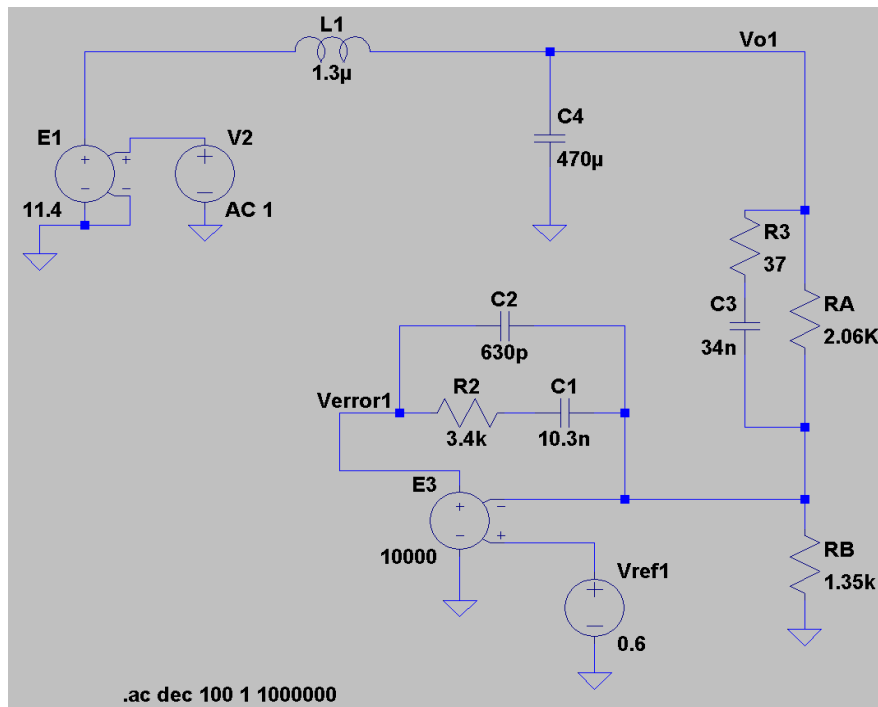


Figura 8: esquemático en LT Spice.

La primera simulación se realizó con los datos calculados de las fórmulas anteriores, lo cual da como resultado un margen de fase bajo ( $\sim 28^\circ$ ).

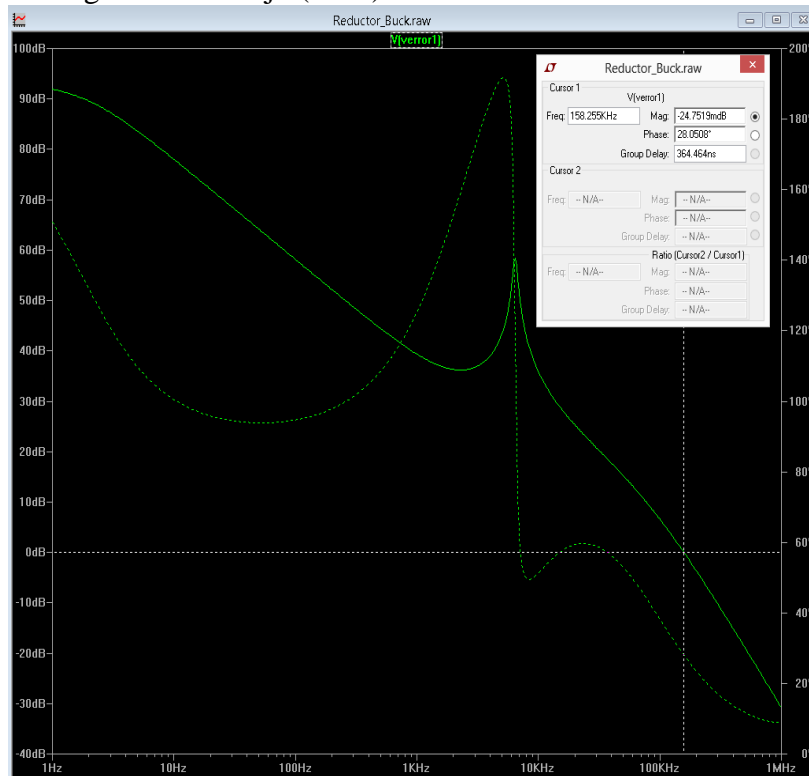


Figura 9: simulación de AC del regulador con datos calculados.

Posteriormente, ajustando los valores de  $C3 = 22\text{nF}$ ,  $R3 = 45\ \Omega$ , se logra obtener un margen de fase de  $\sim 57^\circ$ .

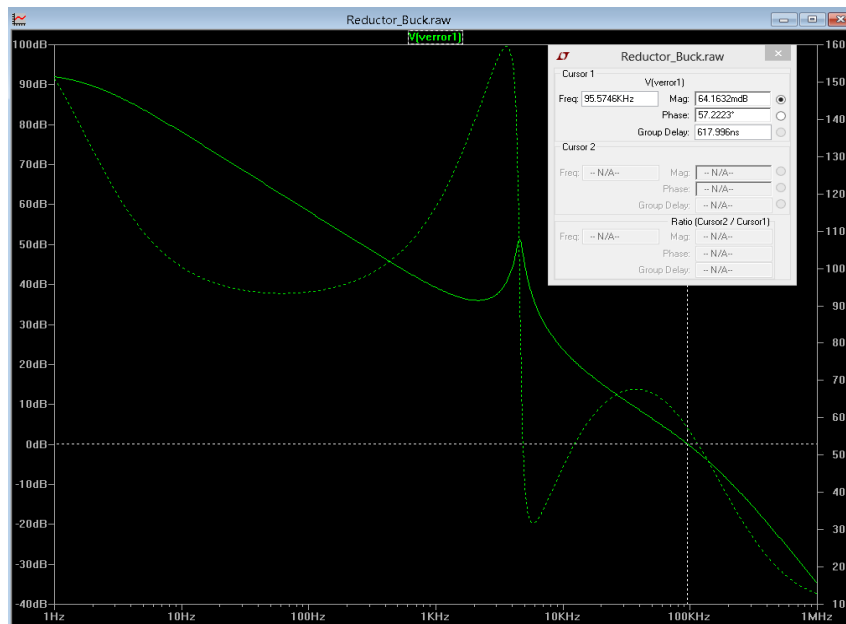


Figura 10: simulación de AC del regulador después del ajuste.

La simulación del transitorio se observa en la siguiente imagen, en donde la salida se muestra con la línea azul, teniendo a la salida 1.5V, con un Istep de 7A y un voltaje Pk-Pk de ~28mV.

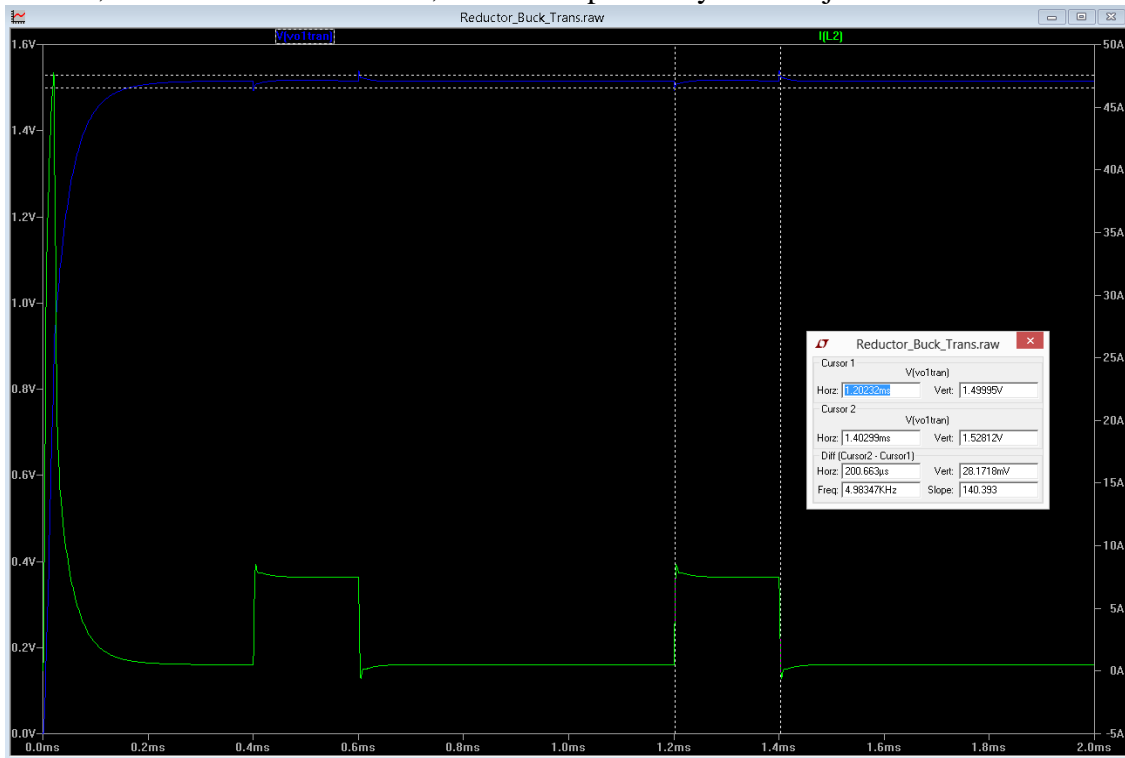


Figura 11: voltaje a la salida del regulador.

Así también, se implementó el esquemático del diseño utilizando la herramienta de Orcad, el cual se muestra en la siguiente figura.

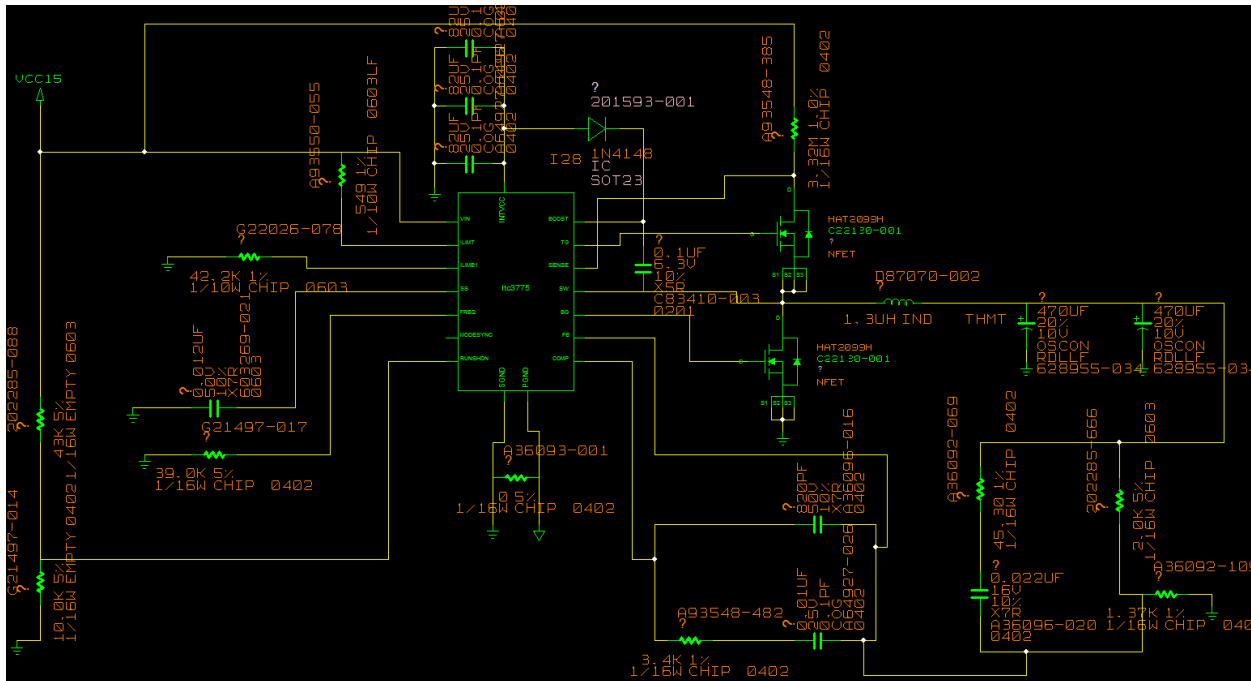


Figura 12: esquemático del regulador de voltaje

### Costos y materiales.

Finalmente, se hizo la búsqueda de componentes que conforman la implementación del regulador. Cabe mencionar que esto se hizo sólo con el propósito de tener una idea sobre el valor aproximado del diseño, sin llegar a hacer el diseño físicamente. La siguiente tabla muestra el fabricante, número de parte, descripción, cantidad, precio y proveedor de cada componente utilizado en el diseño.

Fabricante	Part No.	Descripción	Qty	Precio	Consult	Stock	Proveedor
Nichicon	UPW1V820MED	CAP ALUM 82UF 35V 20% RADIAL	3	\$0.99	23/11/14	6054	Digikey
Nichicon	UMV1H0R1MFD1TP	CAP ALUM 0.1UF 50V 20% RADIAL	1	\$0.12	23/11/14	2000	Digikey
Panasonic	ECA-1EM471B	CAP ALUM 470UF 25V 20% RADIAL	2	\$0.28	23/11/14	27000	Digikey
Murata Electronics	GRM155R71E103KA01D	CAP CER 10000PF 25V 10% X7R 0402	2	\$0.2	23/11/14	3623671	Digikey
Murata Electronics	GRM155R71C223KA01D	CAP CER 0.022UF 16V 10% X7R 0402	1	\$0.1	23/11/14	391096	Digikey

Murata Electronics	GRM033R71E821KA01D	CAP CER 820PF 25V 10% X7R 0201	1	\$0.1	23/11/14	93404	Digikey
Panasonic	ERJ-1TYF271U	RES 270 OHM 1W 1% 2512 SMD	2	\$1.92	23/11/14	26800	Digikey
Panasonic	ERJ-3EKF45R3V	RES 45.3 OHM 1/10W 1% 0603	1	\$0.1	23/11/14	88130	Digikey
Panasonic	ERJ-2GE0R00X	RES 0.0 OHM 1/10W JUMP 0402	1	\$0.1	23/11/14	2795572	Digikey
Samsung	RC1005F2102CS	RES 21K OHM 1/16W 1% 0402	2	\$0.2	23/11/14	108225	Digikey
Rohm	MCR03ERTF3902	RES 39K OHM 1/10W 1% 0603	1	\$0.1	23/11/14	23970	Digikey
Panasonic	ERJ-3EKF4322V	RES 43.2K OHM 1/10W 1% 0603	1	\$0.1	23/11/14	77136	Digikey
Rohm	MCR01MRTF1002	RES 10K OHM 1/16W 1% 0402	1	\$0.1	23/11/14	723759	Digikey
Panasonic	ERJ-3EKF3401V	RES 3.4K OHM 1/10W 1% 0603	1	\$0.1	23/11/14	63010	Digikey
Panasonic	ERJ-3EKF1371V	RES 1.37K OHM 1/10W 1% 0603	1	\$0.1	23/11/14	46866	Digikey
Panasonic	ERJ-2RKF2051X	RES 2.05K OHM 1/10W 1% 0402	1	\$0.1	23/11/14	34883	Digikey
Panasonic	ERJ-3GEYJ305V	RES 3M OHM 1/10W 5% 0603	1	\$0.1	23/11/14	85465	Digikey
Micro Commercial Co	1N4742A-TP	DIODE ZENER 12V 1W DO41	1	\$0.25	23/11/14	14765	Digikey
Murata Power Solutions Inc	38S132C	FIXED IND 1.3UH 10.5A 5 MOHM	1	\$1.77	23/11/14	1250	Digikey
Renesas Electronics	RJK0305DPB-00#J0	MOSFET N-CH 30V 30A LFPK	2	\$3.04	23/11/14	5909	Digikey
Linear Technology	LTC3775EMSE#PBF	IC REG CTRLR BUCK PWM VM 16-MSOP	1	\$3.82	23/11/14	331	Digikey

Costo total del diseño: \$13.69 USD.

## **B. MÓDULO DE CONTROL DE CARROCERÍA AUTOMOTRIZ**

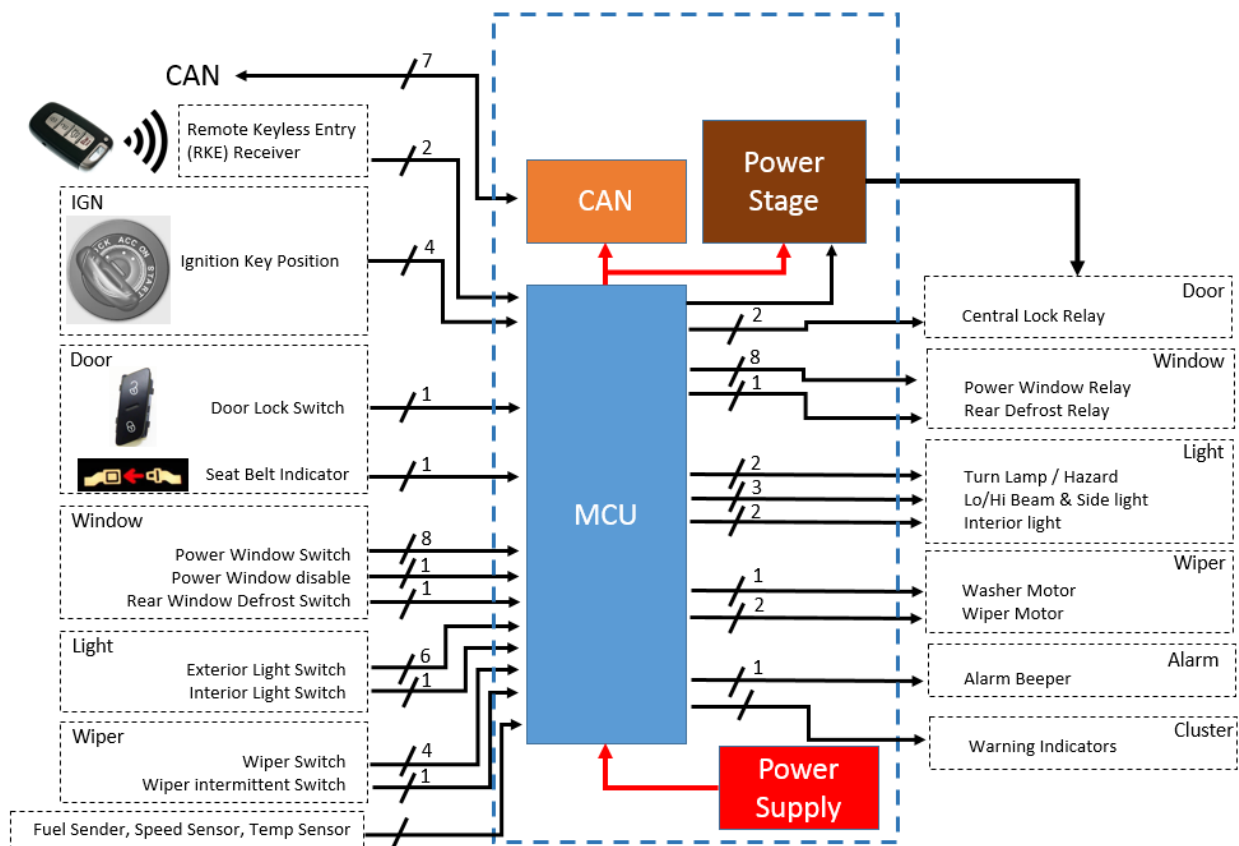
Contenido

<b>Introduction</b> .....	<b>28</b>
<b>1. System Boundary Diagram</b> .....	<b>28</b>
<b>2. P-Diagram</b> .....	<b>29</b>
<b>3. Functional Process Map</b> .....	<b>30</b>
<b>3. Introduction</b> .....	<b>31</b>
<b>4. Software Architecture</b> .....	<b>31</b>
i. Application Layer (APL).....	32
ii. ECU & MCU Abstraction Layer.....	32
iii. System Services Layer.....	33
iv. Hardware Layer.....	34
<b>5. Software Design</b> .....	<b>34</b>
v. <i>Scheduler</i> .....	35
vi. CAN driver.....	39
vii. PWM driver.....	39
viii. ADC driver.....	40
ix. GPIO driver.....	42
x. PIT Driver.....	42
xi. Watchdog driver.....	43
xii. CAN Handler.....	43
xiii. ADC Handler.....	45
<b>6. Customer and Technical Requirements</b> .....	<b>51</b>
<b>7. House of Quality</b> .....	<b>55</b>
<b>8. Failure Modes and Effects Analysis</b> .....	<b>56</b>

## Introduction

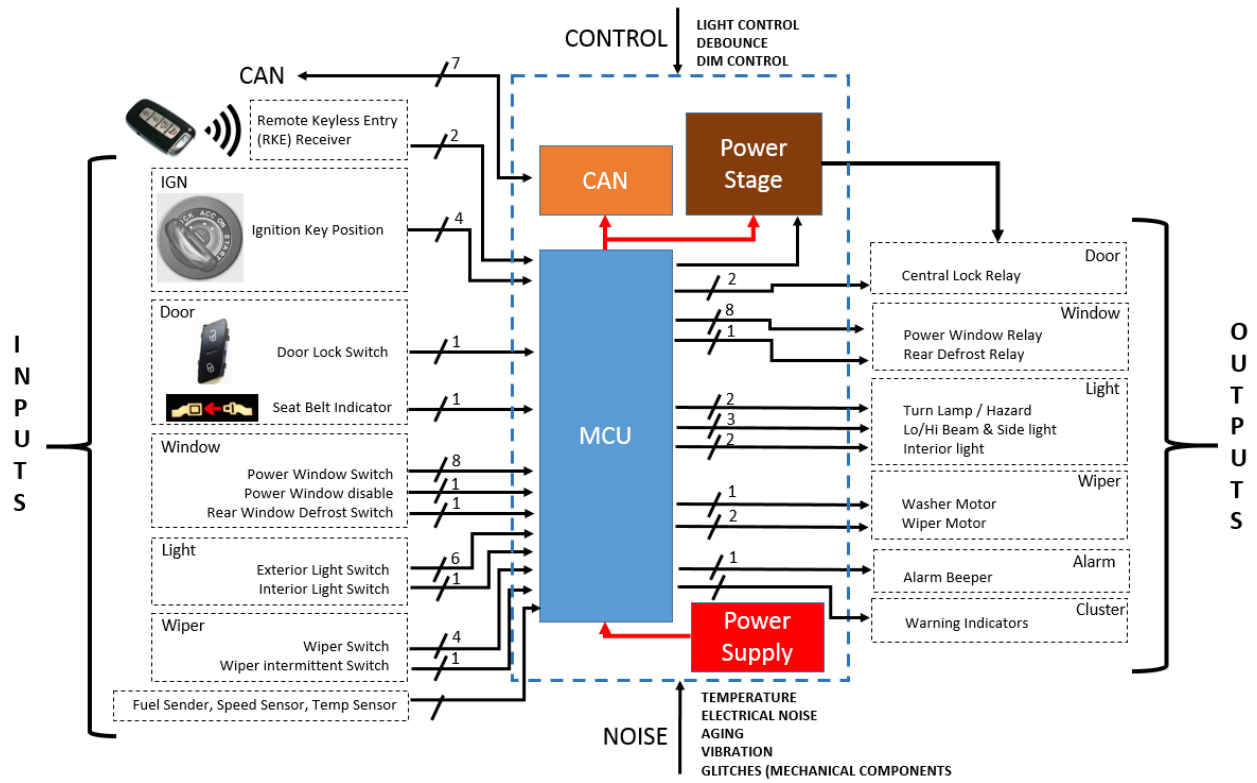
This document describes the system architecture of the Body Control Module, it contains the System Boundary diagram, Parameter diagram and Functional Process map.

### 1. System Boundary Diagram

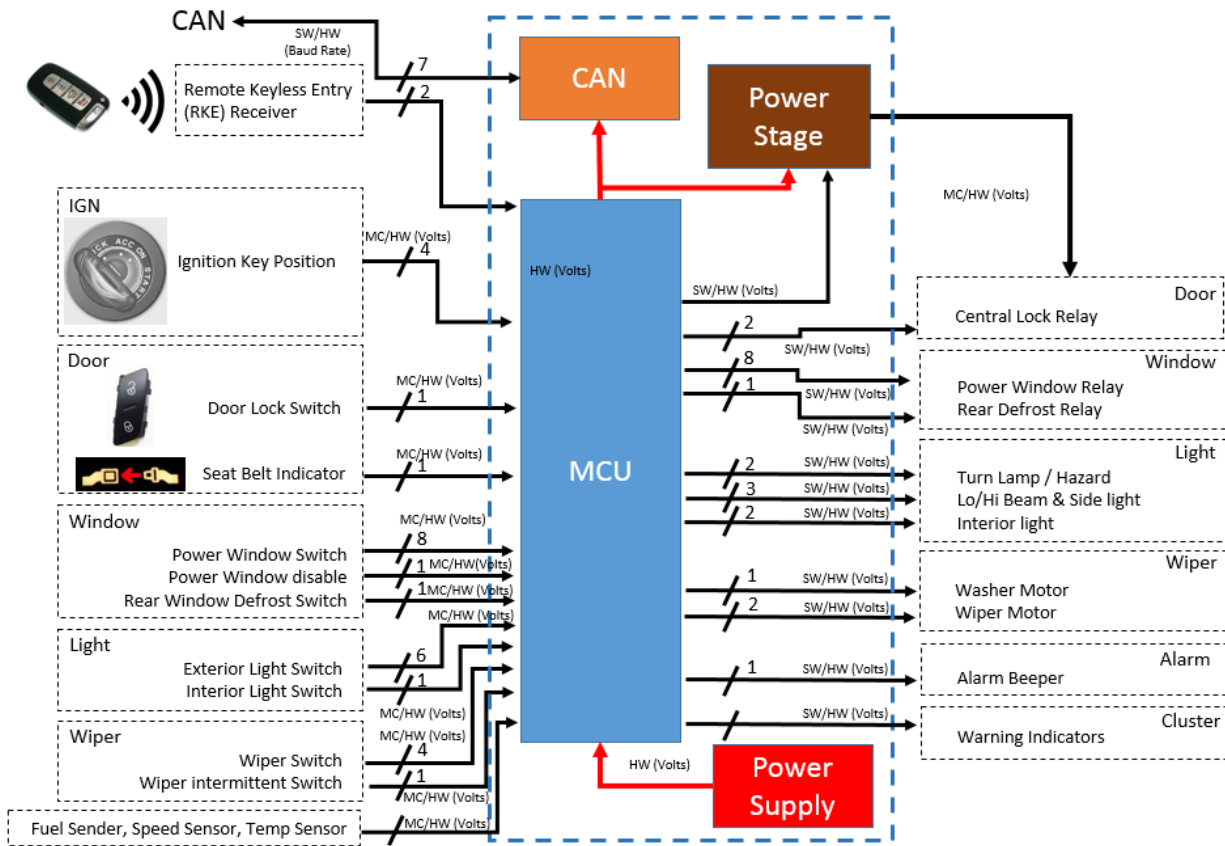




## 2. P-Diagram



### 3. Functional Process Map



### 3. Introduction

The current document describes the software architecture of the BCM (Body Control Module). For its implementation the AUTOSAR (Automotive Open System Architecture) model will be used as the main architecture reference; however, it won't be fully compliant with AUTOSAR since only some layers will be used.

### 4. Software Architecture

The software architecture model uses the system requirements, the software requirements and the layers suggested by the AUTOSAR model to generate the architecture.

For the current implementation of the software architecture the following layers will be included:

- **Application Layer**
- **ECU and MCU Abstraction Layer**
- **System services Layer**
- **Hardware Layer**

Each of these layers contain different blocks for the implementation as shown in the diagram of the *Figure 1*. In some cases, the blocks can expand into two different layers: e.g. System Services Module.

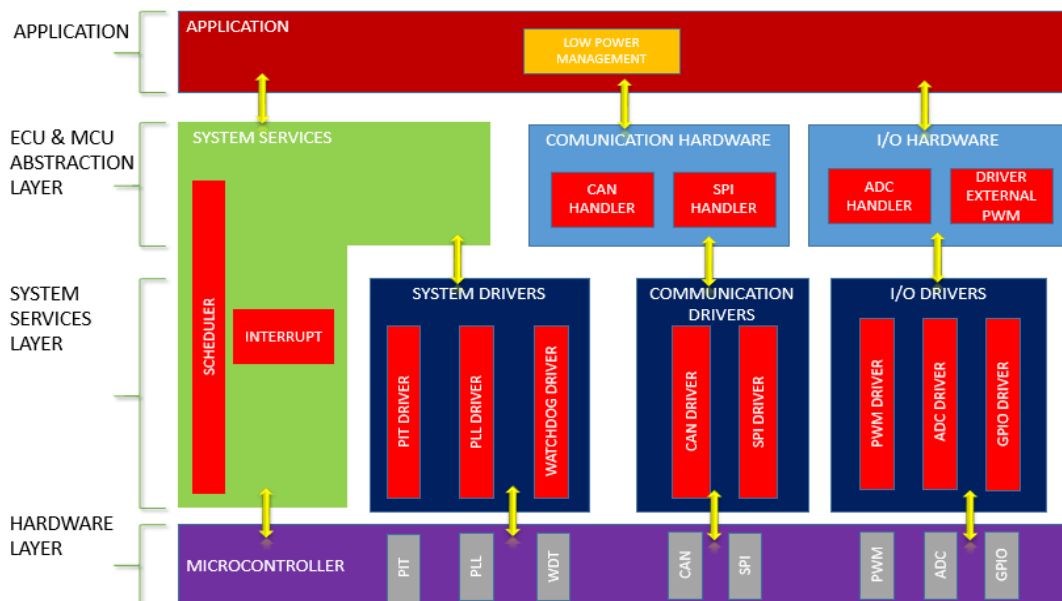


Figure 1: BCM Software Architecture

### **i. Application Layer (APL)**

This layer will be in charge of the low power management, which can switch the operation mode of the BCM to or from low consumption mode.

When the system detects a low voltage condition, the BCM turns off the board and interrupts the CAN communication after sending a message to the cluster about the battery status.

### **ii. ECU & MCU Abstraction Layer**

The MCU layer contains software modules with direct access to the microcontroller internal peripherals and memory mapped external devices. Their main function is to make higher software layers independent of microcontroller.

The ECU layer provides an API for access to peripherals and devices regardless of their location (microcontroller internal/external), the number of existing devices of that type, the hardware realization of the same devices and their connection to the microcontroller. These interfaces contain the functionality to abstract the hardware realization of any specific device and they don't change the content of the data.

On the same line, the handlers are specific interfaces that control the concurrent, multiple and asynchronous access of one or multiple clients to one or more drivers without changing the content of the data.

### **System Services (OS)**

This block includes basic hardware configuration consisting of an Interrupt Handler module and a OS scheduler module. The named modules have direct access to the microcontroller internal peripherals and memory mapped external devices. Their main function is to make higher software layers independent of microcontroller.

**OS Scheduler:** This module is responsible for managing the CPU resources' usage, some of the characteristics include: To provide a priority-based scheduling for the CPU tasks, configuration and escalation based on statistic data and sensitive to real-time performance reasoning.

**Interrupt Handler:** The module responsible to handle the system interrupts. These are classified in three categories: Periodic Interrupts, Communication Interrupts and I/O Interrupts

### **Communication Hardware Abstraction**

This group of modules interfaces the communication controllers and the ECU hardware layout. For all communication systems, a specific communication hardware abstraction is

required. Its task is to provide equal mechanism to access a bus channel regardless of its location (on-chip/ on-board).

### **I/O Hardware Abstraction**

This group of modules interface the peripheral I/O devices (on-chip or onboard) and the ECU hardware layout (excluding sensors and actuators). The different I/O devices are accessed via an I/O signal interface. Its task is to represent I/O signals as they are connected to the ECU hardware (e.g. current, voltage and frequency measurements).

#### **iii. System Services Layer**

The drivers will provide a software interface to the hardware devices, enabling the operating systems and the application layer to access hardware functions without needing to know precise details of the hardware being used.

#### **Input/output drivers**

The BCM will interact with sensors and actuators through the drivers for Analog and Digital Signals. Some of the digital signals will be used as luminosity signals (LEDs) to indicate status of the system.

#### **System drivers**

These drivers allow the execution of the scheduler, take over the execution flow, verify the correct operation of the system (through the watchdog), verify the integrity of the memory and provide the OS tick for the scheduler time base.

#### **Communication drivers**

These drivers will be responsible for handling the communication Framework, the CAN and SPI interfaces and the I/O and Network management.

#### iv. **Hardware Layer**

Access to the hardware is routed through the hardware layer in order to prevent direct access to the microcontroller registers from higher-level software. The hardware layer ensures a standard interface to the different components of the ECU Software Layer. Its main function is to manage the MCU peripherals and provide hardware values to the ECU modules in the layers above. The module will do this through a notification structure to support the distribution of commands, responses and information to the different processes. The peripherals handled by the hardware layer are:

- Digital I/O (DIO)
- Analog/Digital Converter (ADC)
- Pulse Width (De)Modulator (PWM, PWD)
- Capture Compare Unit (CCU)
- Watchdog Timer (WDT)
- Serial Peripheral Interface (SPI)
- Controller Area Network Bus (CAN)

## 5. **Software Design**

The BCM software design represents an architectural structure with interfaces that simplify the connections between the different modules and the external environment.

The life cycle of the BCM development is described next:

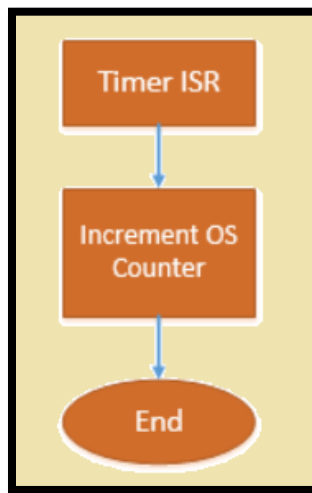
1. **Requirements:** Identify the problem and classify the requirements into functional and non-functional.

2. **Model of problem:** Analyze and build the model of the problem by taking the AUTOSAR reference as the base line.
3. **Postulate a design solution:** Organize the elements and modules to build the software architecture.
4. **Validate Solution:** Generate the V-Model to validate the design.
5. **Refine Design Solution:** Iterate through the software design cycle to implement corrections and improvements to the model.
6. **Implement Solution:** Programming the software design in C language using Atmel IDE.

#### v. Scheduler

The OS scheduler uses a real time periodic interrupt implemented by hardware in order to generate the heartbeat of the operative system, this heartbeat is known as the OS tick.

The flow of interrupt service routine (ISR) of the timer interrupt is described in the *Figure 13*. The timer interrupt must be set before the configuration of the PLL.



*Figure 13: ISR of the periodic timer interrupt.*

The periodic operation of the BCM is described in the task tables. A brief description, the periodicity and the priority of the tasks to be executed are described in the task tables. A fully cooperative scheduling algorithm is used to schedule the execution of the all task in the system. The current implementation offers flexibility to configure different modes of operation for the

BCM. The modes of operation can be switched to *normal mode* and *low voltage mode* and are represented in two different tables. *Table 1* shows the task tables definition.

*Table 1: Task table definitions*

<b>Task Name</b>	<b>Normal mode Functions</b>	<b>Low power mode Functions</b>	<b>Rate [mS]</b>
<b>Task2MS</b>	<i>Debounce Key Status Debounce Brake Pedal</i>		<i>2</i>
<b>Task4MS</b>	<i>Get Battery Level Get Brake Pedal Status</i>	<i>Debounce Key Status Debounce Brake Pedal Get Brake Pedal Status</i>	<i>4</i>
<b>Task8MS</b>	<i>Set PWM Duty Cycle Send Data to Cluster Get Lights Signal Status Get Stalk Status</i>	<i>Send Data to Cluster Get Lights Signal Status Get Stalk Status</i>	<i>8</i>
<b>Task16MS</b>	<i>Process Received CAN Frames</i>	<i>Process Received CAN Frames</i>	<i>16</i>
<b>Task32MS</b>	<i>Transmit Stalk Status Transmit Lights Status Transmit Brake Pedal Status</i>	<i>Set PWM Duty Cycle Transmit Stalk Status Transmit Lights Status Transmit Brake Pedal Status</i>	<i>32</i>

After hardware configuration and prior to the execution of the tasks, the real time scheduler needs to find the active task table, validate that the table is the correct one and store the address of the valid table for the Kernel. This procedure is represented in the flow diagram of the *Figure 14*.



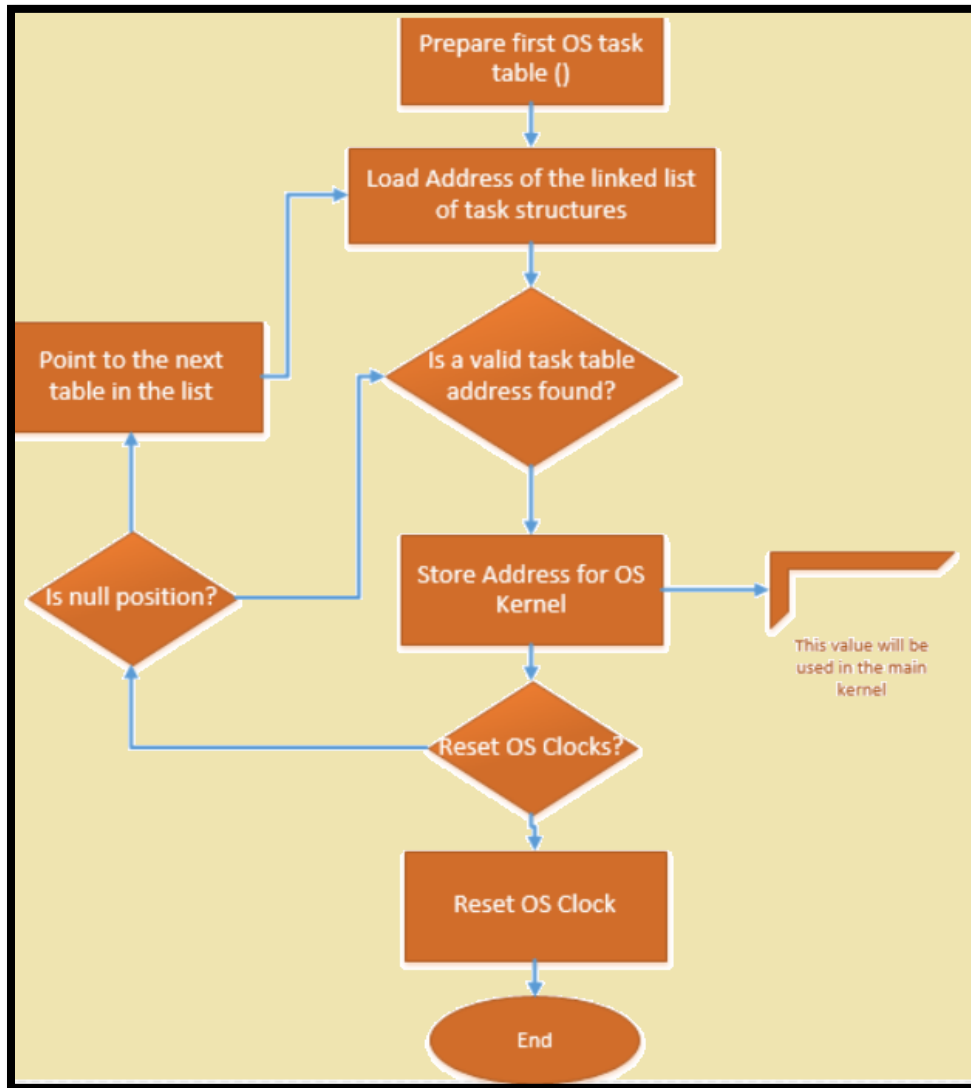
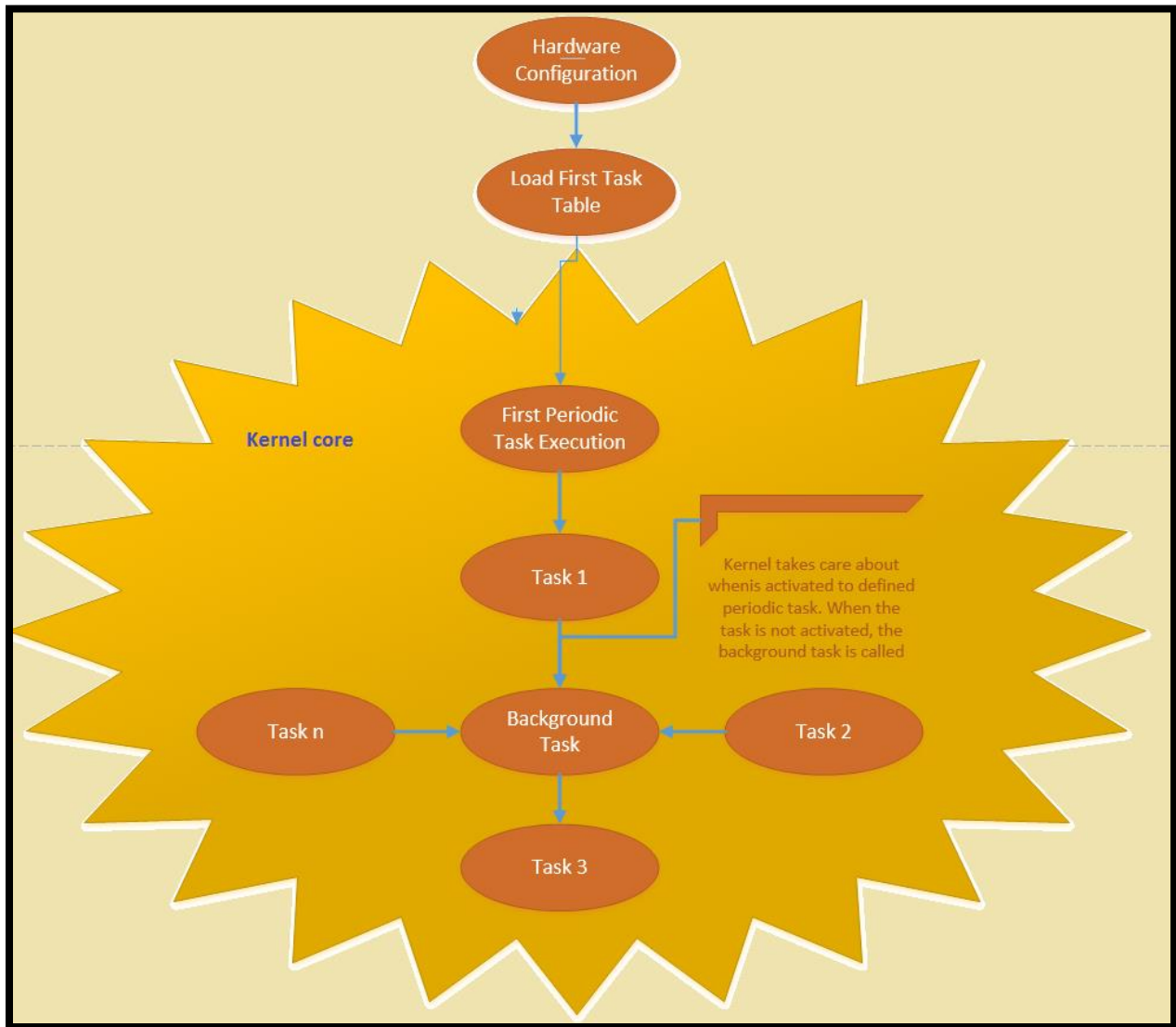


Figure 14: Flow Diagram of the preparation of the first OS task table.

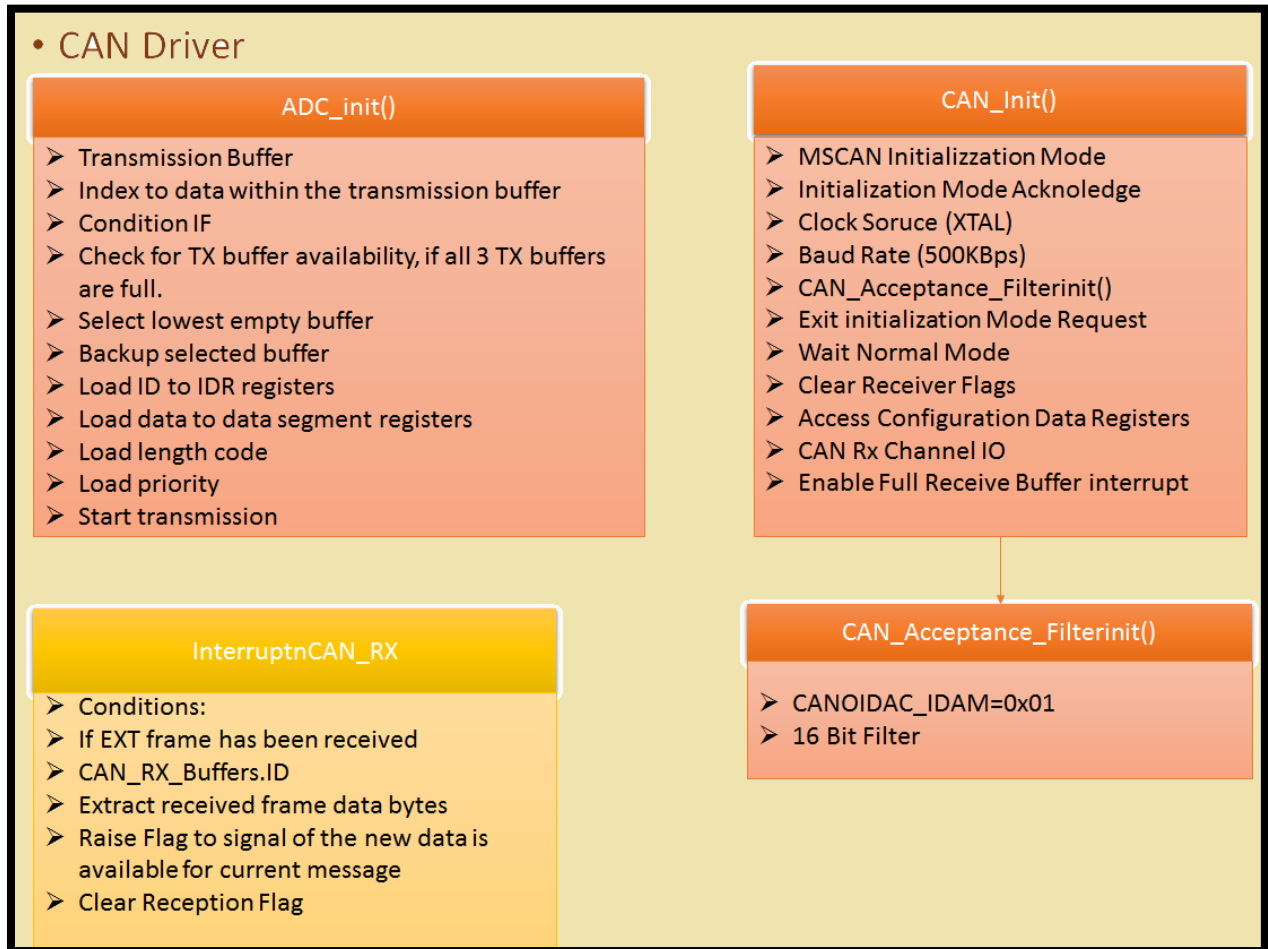
The decision of which task will be executed will be performed by the kernel of the scheduler. The logic implementation of this module is illustrated in *Figure 15*.



*Figure 15: OS kernel block diagram.*

**vi. CAN driver.**

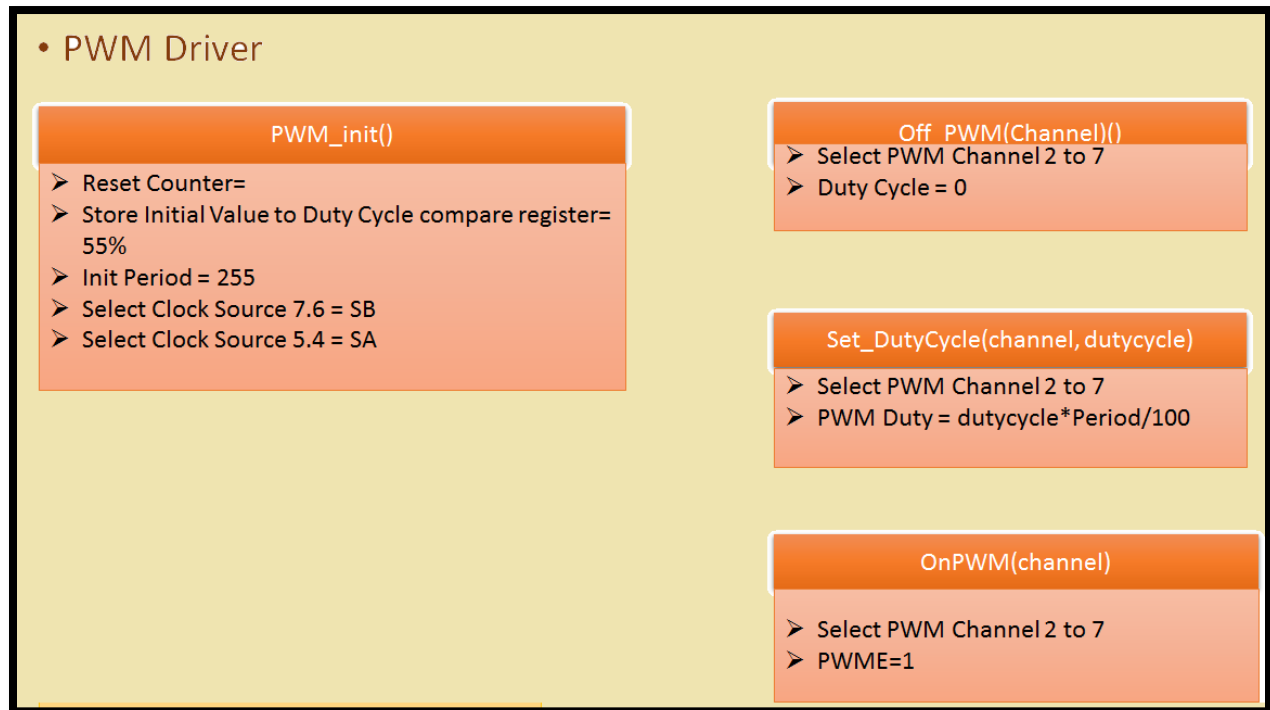
CAN Bus is a vehicle bus standard designed to allow microcontrollers and devices to communicate with each other a vehicle without a host computer. It is also a message-based-protocol, designed specifically for automotive applications. The CAN bus driver allows communication between BCM and Cluster for sending and receiving messages, *Figure 16*.



*Figure 16: CAN block diagram.*

**vii. PWM driver.**

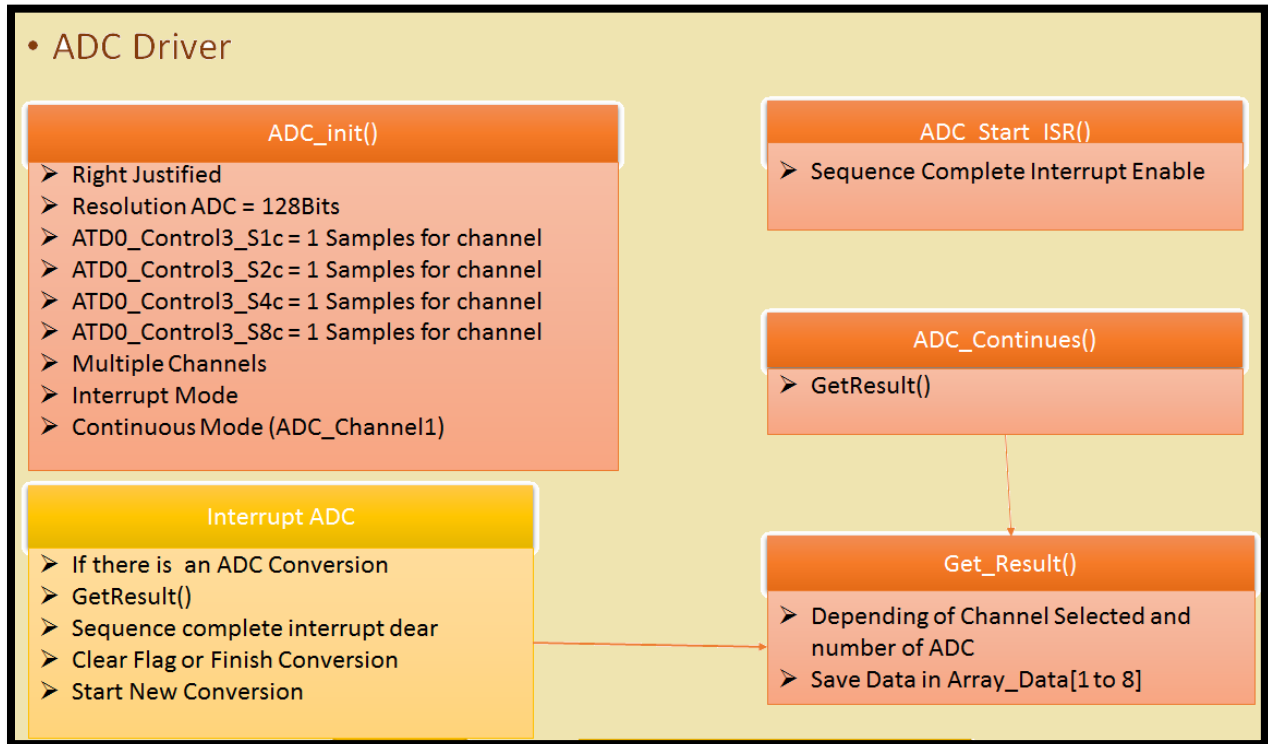
The pwm driver is used to control the intensity of the light (both external and internal). The BCM system has pwm signals that are connected into the input of the power switch, this driver has four functions that are shown in *Figure 17*.



*Figure 17: PWM block diagram.*

### viii. ADC driver.

The majority of the input sensors to the BCM are analog, the signals need to be digitalized by using an ADC port. The ADC can be configured to 10 or 12 bit resolution. For the current implementation, the ADC will be configured to 12 bits interrupt mode conversion. The sensors that are connected to the ADC ports are: temperature, humidity, speed, voltage battery and proximity. The functions contained in the ADC driver are shown in *Figure 18*.



*Figure 18: ADC block diagram.*

## ix. GPIO driver.

The GPIO port will be used to indicate the status in the system performance. The BCM will report if the CAN bus lost the communication or if it is working in low power mode by using a LED connected to a GPIO port. The function to use a port are shown in *Figure 19*.



Figure 19: GPIO block diagram.

## x. PIT Driver.

The PIT driver will assist the BCM scheduler to execute the periodic tasks needed by the system. The functions are listed in *Figure 20*.

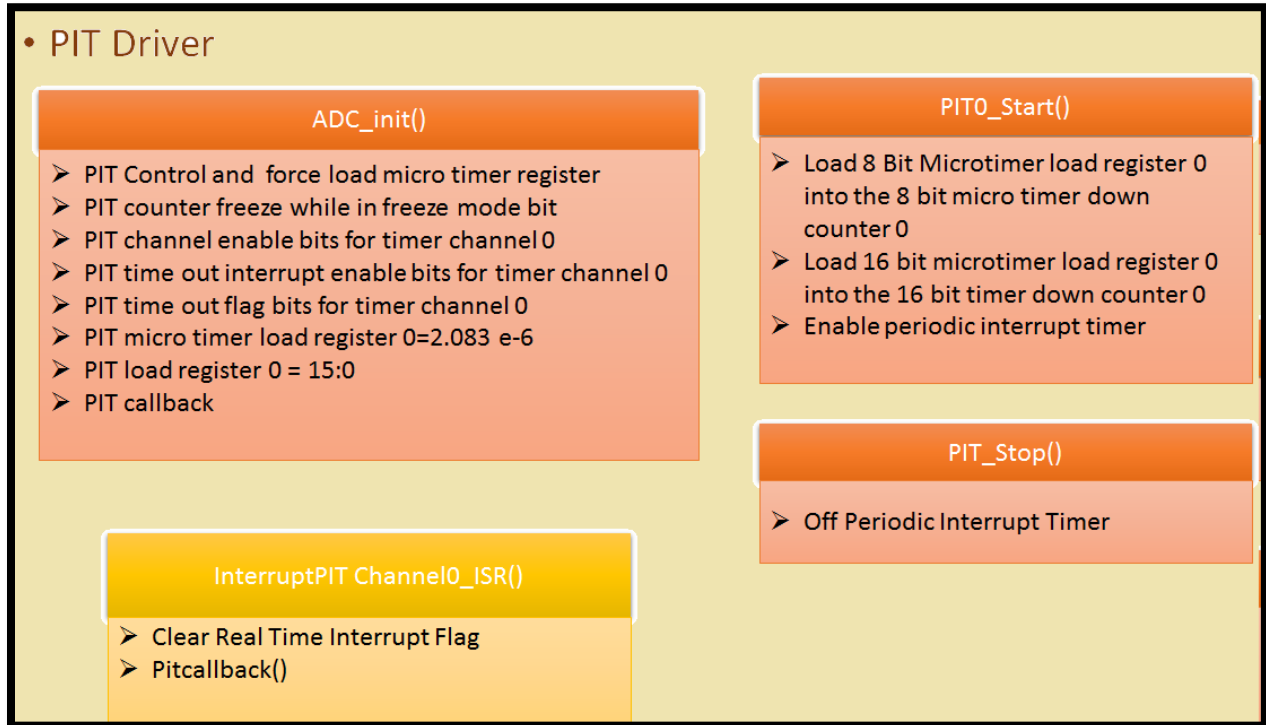


Figure 20: PIT block diagram.

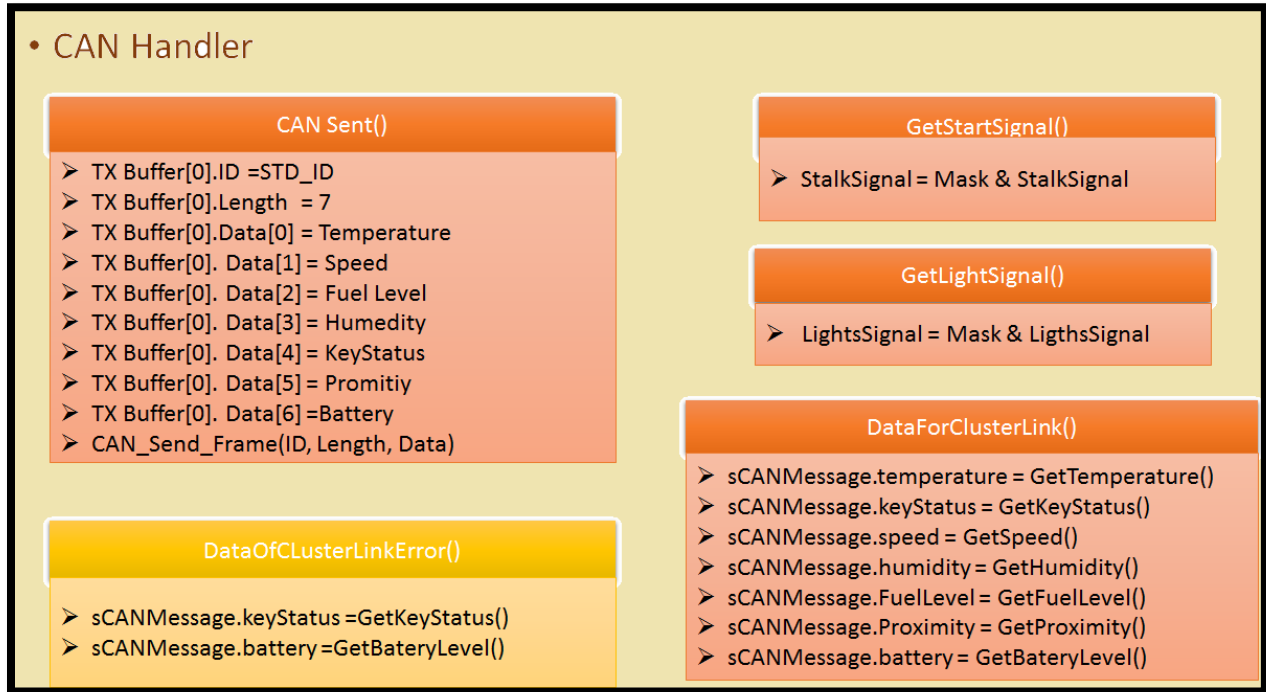
### **xi. Watchdog driver.**

The purpose of the watchdog is to provide the system with a way to recover in the case of incorrect code execution or other events that may cause uncontrolled operation of the MCU. Typically, a watchdog is a continuously running timer that expires or rolls over at a predetermined time interval. This interval is usually determined by the system clock frequency and a watchdog timeout value that is set by the application.

The application must perform some specific action before the timer expires, which causes a reset of the watchdog timer, and a restart of the timeout count. The WD must have the same time value that the task with the less execution time, in this case is 2ms.

### **xii. CAN Handler.**

The CAN Handler contains some functions that permit to obtain the signals of different sensors sending by cluster and send messages with the system status, *Figure 21* has these functions.

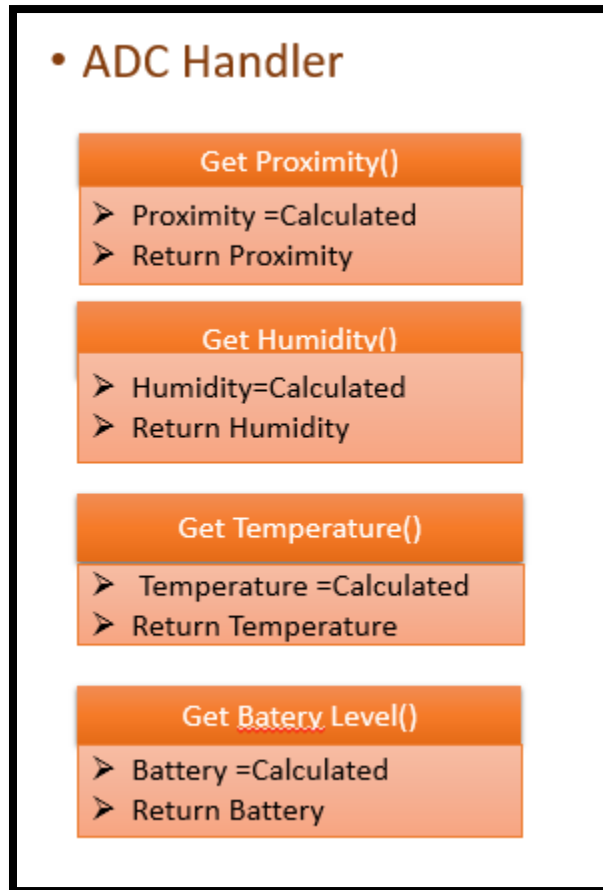


*Figure 21: CAN Handler Block Diagram.*



### xiii. ADC Handler.

It permits to get the values for Proximity, Humidity, Temperature and Battery Level sensors, the *Figure 22* shows the functions that are using in this driver.



*Figure 22: ADC handler block diagram.*

The following diagrams show the algorithm to perform a conversion from a digital value to a number “readable” by the final user, for instance degrees, RH%, km/h, these values are then send to the cluster display. These functions run in a low priority task.

**Temperature:** The temperature sensor has a resolution of 10mV per degree (Celsius). The ADC resolution is 12 bits (max count = 4095), and the CONSTANT is set to 500.

When the temperature sensor reads 30C, it produces a 300mV DC level at its output. When passed through the ADC this value is converted to 250. By applying the following formula, the final value to be displayed in the cluster reads 30C.

$$Cent = \frac{ADCValue * CONSTCENT}{RESOLUTION} = \frac{250 * 500}{4095} = \frac{1250000}{4095} = 30.52^{\circ}C$$

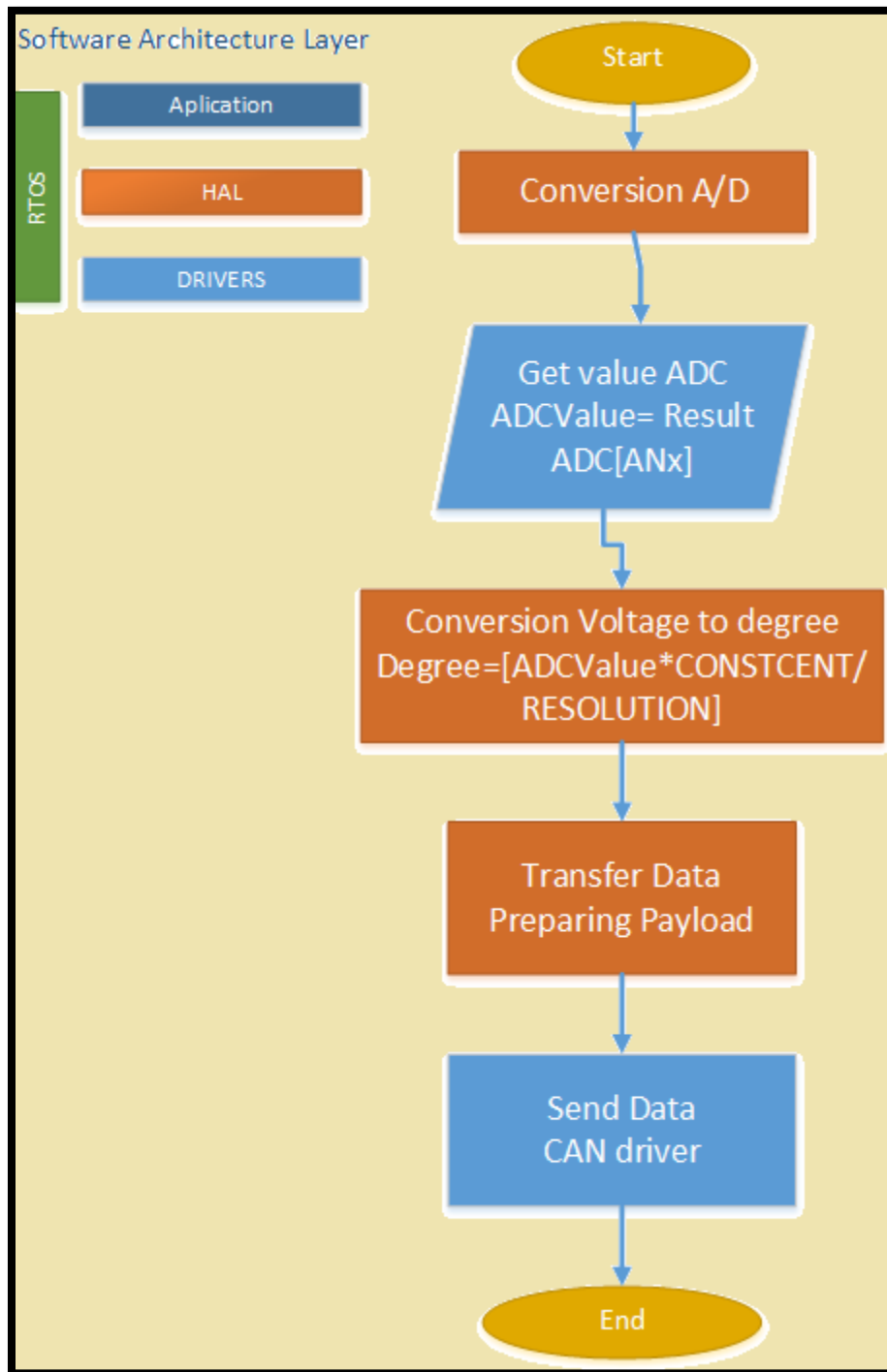


Figure 23: Converter ADC Digital Value to Centigrade.

**Humidity:** The humidity range (10 %RH to 90 %RH) is translated to a voltage range (1 to 2.9V) by the humidity sensor. The following formula is used to calculate the final value to be displayed in the cluster, when considering the maximum range of the sensor and a given constant.

$$HUM = \frac{ADCValue * MAX_PORC}{MAX_RANGE} = \frac{1666 * 90}{2497} = 60.04 \%RH$$

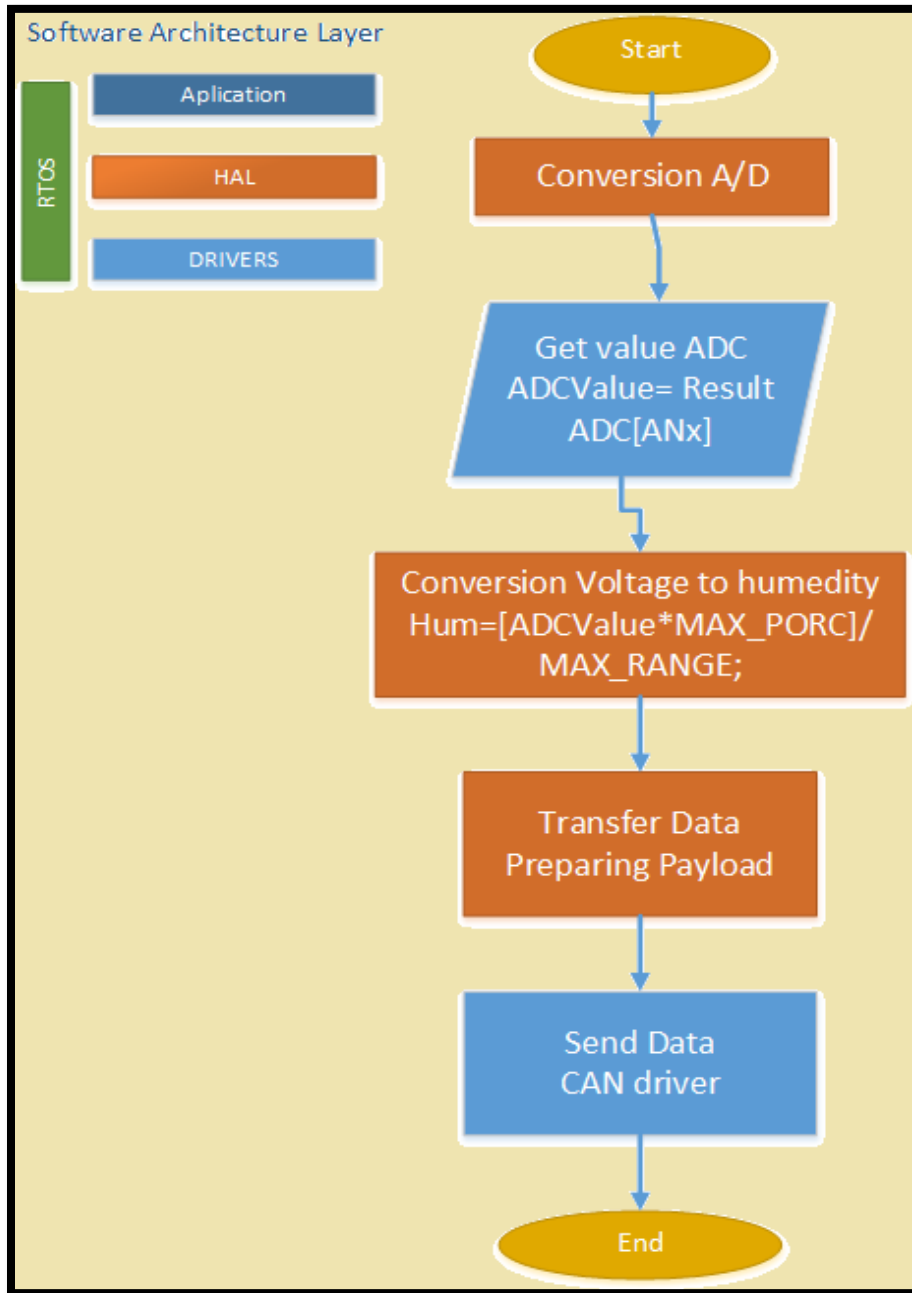


Figure 24: Converter ADC Digital Value to Humidity

**Battery Voltage:** The CONSTVOLT constant is obtained by dividing the ADC resolution by the max battery voltage (12V). If we assume a 4.5V reading in the output, the battery voltage is 11V.

$$Voltage = \frac{ADCValue}{CONSTVOLT} = \frac{3750}{341} = 10.99v.$$

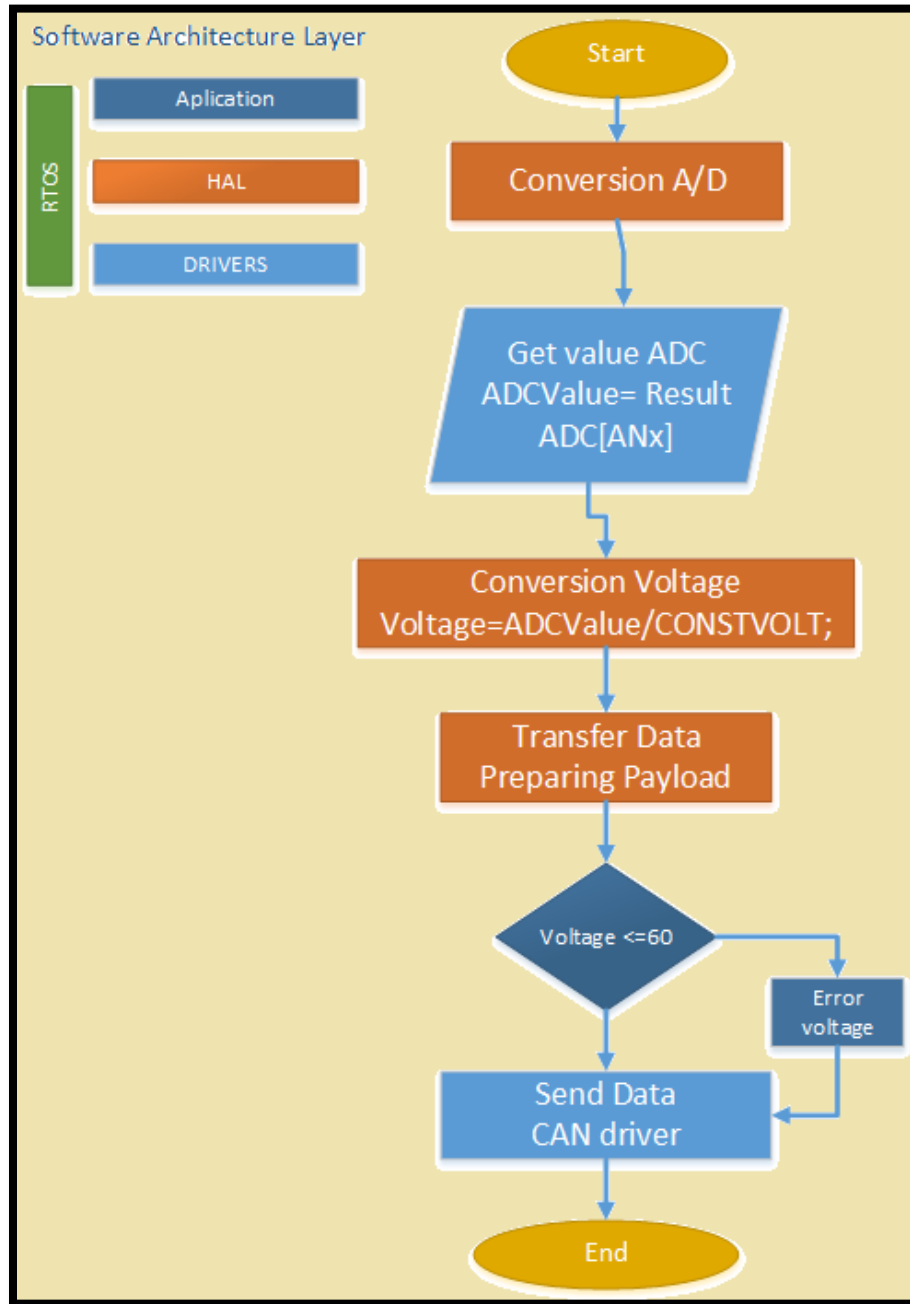


Figure 25: Converter ADC Value to Voltage.

**Proximity:** The proximity sensor resolution is given by:

$$Res_{Sensor} = \frac{5}{512} = 9.8mv = 1 \text{ in or } 2.54cm$$

And the ADC resolution is:

$$Res\_ADC = \frac{5}{4095} = 1.22mv$$

If Res\_Sensor is divided by Res\_ADC the result is 8, which means, each 8 units is equal to 2.54cm. The RESOLUTION\_IN\_CM constant is Res\_Sensor multiplied by 100, at the end of the operation the result is divided by 100 to get the value in centimeters.

$$Dist = \frac{\left(\frac{ADCValue}{PRS\_TO\_ADC\_UNIT}\right) * RESOLUTION\_IN\_CM}{GO\_TO\_CM} = \frac{\left(\frac{1280}{8}\right) * 254}{100} = 398.14cm$$

When an object is present at 4m distance (400 cm), the output voltage is 1.53 or 1280 counts of the ADC.

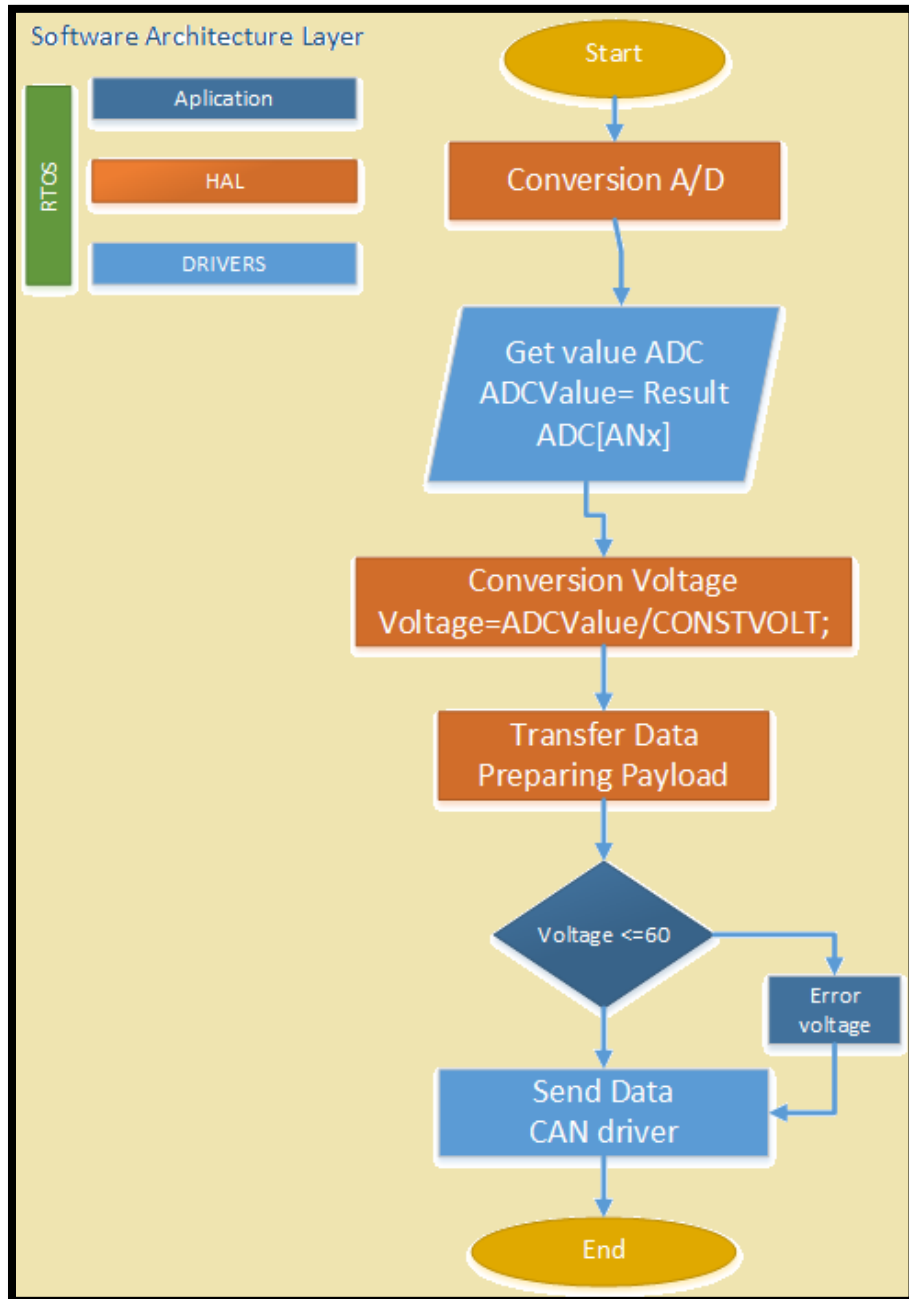


Figure 26: Converter ADC to Distances.

## 6. Customer and Technical Requirements

### Specifications

	Vcc	Icc
<b>Turn Off</b>	13.0v	0.1A
<b>Turn On</b>	13v	2A
<b>Model</b>	Audi A4 2013, zentrale Komfortelektronik H17/s0012 tfk 8E0 959 433 Db	

Table 1. Specifications

### Key words & ID

To start writing requirements is needed to use key words to indicate the importance grade to be implemented. The list are the following:

Key word	Action
<b>Suggestion</b>	Could
<b>Intent</b>	Will/would
<b>Wish</b>	Should
<b>Duty</b>	Shall/have/must

Table2. Key words & actions

To identify each requirement is needed to use an ID or enumerate it with numbers or letters. In this case is used incremental numbers CR (customer requirement) and TR (Technical Requirements).

### Requirements

According the BCM customer requirements are the following using the actions/ID:

Action ID	Description
<b>CR1</b>	The system shall report the engine temperature.
<b>CR2</b>	The system shall report the vehicle speed.
<b>CR3</b>	The system shall report the fuel level.
<b>CR4</b>	The system shall detect the key status.
<b>CR5</b>	The system shall control ignition key switch position.
<b>CR6</b>	The system shall detect interior and exterior lighting status.

<b>CR7</b>	The system shall report voltage of the battery.
<b>CR8</b>	The system shall have CAN communication.
<b>CR9</b>	The system shall have Parking Brake Warning.
<b>CR10</b>	The system shall have ABS Control Module.
<b>CR11</b>	The system shall have Rear/Center Central Locking Switch.
<b>CR12</b>	The system shall have Central Locking and Anti-theft Alarm System Antenna.
<b>CR13</b>	The system shall have door locks.
<b>CR14</b>	The system shall have seat belt indicator.
<b>CR15</b>	The system shall have wiper switch.
<b>CR16</b>	The system shall have wiper intermittent switch.
<b>CR17</b>	The system shall have power window switch.
<b>CR18</b>	The system shall have power window disable.
<b>CR19</b>	The system shall have rear window defroster switch.

Table 3. Customer Requirements and specifications

Action ID	Description
<b>TR1</b>	The system shall have 2 pines to interact with the light switch and relays: interior & external lighting.
<b>TR2</b>	MCU must be used NXP MPC5606.
<b>TR3</b>	The system shall have 4 analog pines to use like sensors inputs: driver assistance, environment humidity, fuel level, brake pedal.
<b>TR4</b>	All actuators must be isolated from MCU.
<b>TR5</b>	Protocol to be used shall be CAN 2.0 for automotive application.
<b>TR6</b>	In low battery the system should be in standby, just can detect key status, and CAN message are lock.
<b>TR7</b>	Speed value shall be in range of 0 - 220 Km/h.
<b>TR8</b>	Fuel value shall be in range of 0 to 45 Liters.
<b>TR9</b>	The alternator voltage should be measure from 0 to 14 volts.
<b>TR10</b>	The handbrake should be indicated activated or deactivate.



<b>TR11</b>	The system shall have Lock, Accessory, on and start the ignition switch positions.
<b>TR12</b>	The coolant level should be reported low.
<b>TR13</b>	The hood status shall be indicated open or close.
<b>TR14</b>	The brake fluid pressure shall be indicated when have low level.
<b>TR15</b>	The oil pressure should be reported low or high.
<b>TR16</b>	The system shall have a security indicator.
<b>TR17</b>	The door system must have central lock relay.
<b>TR18</b>	The windows shall have power window relay.
<b>TR19</b>	The system must have alarm beeper.
<b>TR20</b>	The system must have washer motor.
<b>TR21</b>	The system must have wiper motor.
<b>TR22</b>	The windows shall have rear defrost relay.
<b>TR23</b>	The system must have hazard switch.
<b>TR24</b>	The system must have turn/hazard lights.
<b>TR25</b>	The system must have fog lights.
<b>TR26</b>	The system shall have warning seat belt indicator light.

Table 4. System Requirements

ECU	Packet Name	signal name	signal function	Measure	pck Length	signal bit	signal byte
200h	BCM_SGN	Temperature	Environment temperature	Celsius	64	1	0
		Speed	Car speed	KM/h		2	0
		Fuel Level	Measure of the fuel tank			3	0
		Humidity	Measure of the environment humidity			4	0
		Key Status	key OFF=0, key ON=1			5	0
		Object Sensing	Measurement of proximity to an object			6	0
		Low voltage Status	Normal Mode=0, low voltage mode=1			7	0
		Battery Voltage	Measurement of battery voltage				

		Reserved	Unused			8	0
--	--	----------	--------	--	--	---	---

Table 5. Transmitter

Another Specifications:

Description		Turn On lights	Light position knob
Turn on	Turn on with rotate a quarter	Cluster	Turn off
Turn signal switch	The user have touch up and blink 3 times to the right. Come back neutral position.	Right BL,R, right sidelight	Turn off
	Touch down and blink 3 times to the left. Come back neutral position.	Left BL,L, left sidelight	Turn off
	Toggle must toward up appear right directional. Toggle come back the original position when the card turn right.	Right BL,R, right sidelight	Turn off
	Toggle shall toward down appear left directional. Toggle come back the original position when the card turn left.	Left BL,L, left sidelight	Turn off
	Toggle toward inside activate high lights, Come back neutral position.	58 (right & left)	Turn off
	Toggle shall toward out and appear high lights and the toggle shall not come back original position.	Nothing	Turn off
	Action 2, but more lights turn on	All actions 2, + TFL, 58, NSL, 58L, 58R, X	sidelight
	Action 3, but more lights turn on	All actions 3, + TFL, 58, NSL, 58L, 58R, X	sidelight
	Action 3, but more lights turn on	All actions 4, + TFL, 58, NSL, 58L, 58R, X	sidelight
	Action 5, but more lights turn on	All actions 5, + TFL, 58, NSL, 58L, 58R, X	sidelight
	Action 6, but more lights turn on	All actions 6, + TFL, 58, NSL, 58L, 58R, X	sidelight
	Action 7, but more lights turn on	All actions 7, + TFL, 58, NSL, 58L, 58R, X	sidelight
	Action 2, but more lights turn on	All actions 8, + 56b	headlights
	Action 3, but more lights turn on	All actions 9, + 56b	headlights
	Action 4, but more lights turn on	All actions 10, + 56b	headlights
	Action 5, but more lights turn on	All actions 11, + 56b	headlights
	Action 6, but more lights turn on	All actions 12, + 56b	headlights

	Action 7, but more lights turn on	All actions 13, + 56b, 56a	headlights
	The same action when have all lights turn on	See action 14	Auto

Table 6. Light position knob

Table 7. Receiver

ID	ECU	Packet Name	signal name	signal function	Status	pck Length	signal bit	signal byte
100h	Cluste r	CLSTR_SG N_LIGHTS	Interior Light	Have actual status of the interior Light signal	0= OFF, 1=ON	2	1	0
			Exterior Light	Have actual status of the exterior Light signal	0= OFF, 1=ON		1	1
		CLSTR_SG N_STALK	Optical Horn	Have actual status of the Optical Horn	0= OFF, 1=ON	5	1	2
			Low Beam	Have actual status of the low Beam signal	0= OFF, 1=ON		1	3
			High Beam	Have actual status of the high Beam signal	0= OFF, 1=ON		1	4
			Turn Right	Have actual status of the turn Right signal	0= OFF, 1=ON		1	5
			Turn Left	Have actual status of the turn Left signal	0= OFF, 1=ON		1	6
			Reserved	Unused	0= OFF, 1=ON	1	1	7

Table 7. Receiver

## 7. House of Quality



HoQ BCM v0p2.xlsx

## 8. Failure Modes and Effects Analysis



FMEA\_IngSW.xlsx

## C. CODIFICADOR DE CANAL EN UN SISTEMA DE COMUNICACIÓN DIGITAL

### Objetivo:

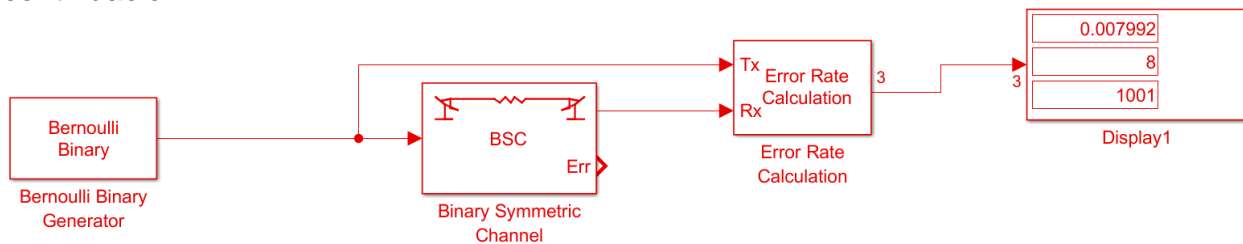
- Comprender el funcionamiento de un codificador de canal en un sistema de comunicación digital.
- Realizar un sistema de comunicación que contenga la función de codificación de canal y comparar los resultados respecto al sistema sin codificador.

### Actividades:

**NOTA:** Este trabajo es para ser realizado con la herramienta de simulación Simulink de MATLAB.

1. Realizar en la plataforma de simulación MATLAB un sistema de comunicación básico como el mostrado en la figura 1. El canal binario simétrico permite configurar directamente la probabilidad de error que se desea sobre los datos binarios a su entrada.

Se realizó el sistema de comunicación básico en simulink como se muestra a continuación:

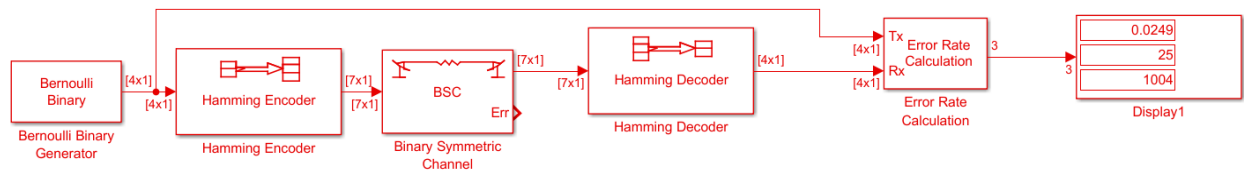


2. Ejecutar la simulación del sistema en varias ocasiones para diversos valores de probabilidad de error del canal, generando por lo menos un total de 1000 bits de información. Utilizar al menos los valores siguientes de probabilidad: 0.005, 0.01, 0.015, 0.025, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3. Constatar que la probabilidad de error obtenida es prácticamente la configurada.

Probabilidad de Error Configurada	Probabilidad de Error Obtenida
0.005	0.007992
0.01	0.01399
0.015	0.01998
0.025	0.03297
0.05	0.05894

0.1	0.1149
0.15	0.1608
0.2	0.2168
0.25	0.2637
0.3	0.3287

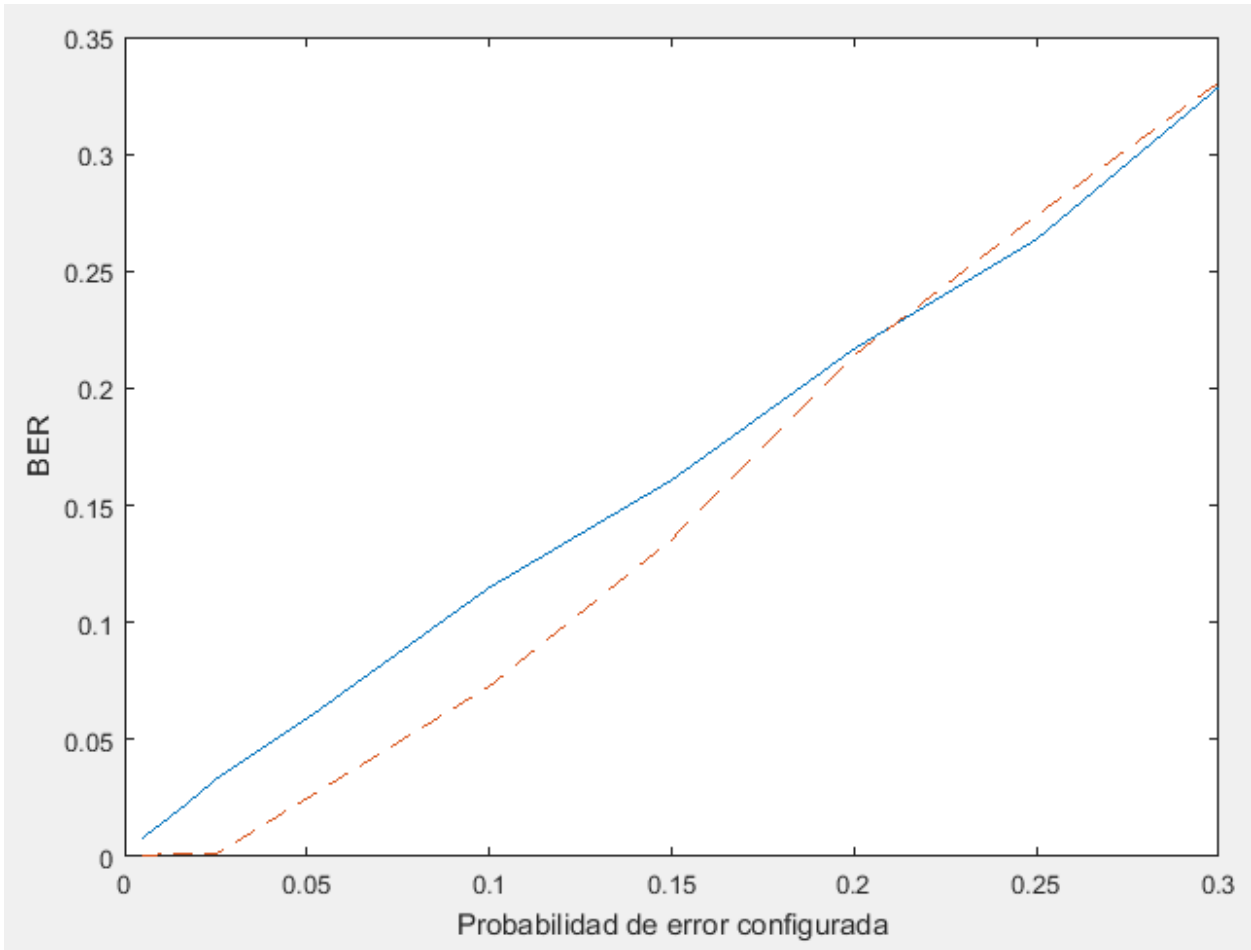
3. Ampliar el sistema de la figura 1 añadiendo un codificador y un decodificador de canal del tipo Codificador de Bloques de Hamming. La configuración por omisión de este bloque es de un código (7,4), figura 2. Es necesario considerar que la entrada al codificador debe ser una trama de la longitud correspondiente a la longitud de la palabra de información (4), por lo que la fuente de información debe de tener configurada su salida de la misma manera (“frame”).



4. Ejecutar varias simulaciones del sistema utilizando los mismos valores de probabilidad de error que los utilizados en el punto 2. Graficar los resultados de los dos casos de ejecución, con y sin codificador de canal (Probabilidad de error configurada en el canal vs. Probabilidad de error obtenida después del decodificador).

Probabilidad de Error Configurada	Probabilidad de Error Obtenida
0.005	0
0.01	0.000996
0.015	0.000996
0.025	0.000996
0.05	0.0249
0.1	0.07271
0.15	0.1355
0.2	0.2141
0.25	0.2739

0.3	0.3307
-----	--------



De la gráfica mostrada arriba se puede observar como el error en la salida disminuyó cuando al sistema se le agregó el codificador Hamming (línea punteada color rojo), pero después llegó a un punto donde incluso tuvo mayor número de errores que el sistema sin codificador (línea azul). Esto es debido a que el código Hamming solo es capaz de detectar y corregir si hubo error en solo un bit.

**5. Analizar el comportamiento del sistema y los resultados, explicando las características particulares del codificador de Hamming. Justificar adecuadamente las conclusiones.**

Se puede observar que la probabilidad de error configurada en el sistema sin decodificador es prácticamente la configurada, esto quiere decir que los datos son afectados directamente por los errores producidos por el ruido en el canal, en cambio en el sistema con codificador de Hamming, los datos codificados son menos

susceptibles a errores que puedan ser generados al transmitirse por el canal ya que ahora la palabra de código contiene los bits de información concatenados a los bits de redundancia, los cuales permiten detectar y corregir los errores en el decodificador bajando así la probabilidad de error obtenida. También se puede observar que la probabilidad de error obtenida, después de haber configurado probabilidades de errores mayores a 0.1, es muy cercada a la configurada y para los casos 0.2 en adelante es mayor que la configurada, esto puede ser debido a las limitantes del codificador Hamming ya que solo puede detectar y corregir un bit, y en estas probabilidades de errores altas puede que exista más de un bit incorrecto.