

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

ESPECIALIDAD EN SISTEMAS EMBEBIDOS



SISTEMA DE CONTROL EMBEBIDO PARA CONVERTIDOR DC/DC DE ALTA ELEVACIÓN

Trabajo final que para obtener el diploma de
ESPECIALISTA EN SISTEMAS EMBEBIDOS

Presenta: Víctor Hugo Huidobro García

Asesor: Dr. Luis Enrique González Jiménez

Asesor: MC Héctor Antonio Rivas Silva

Tlaquepaque, Jalisco, marzo de 2017.

AGRADECIMIENTOS

EL autor agradece sinceramente a las siguientes personas e instituciones:

- A mi esposa Leslie que me ha apoyado desde el principio en esta aventura.
- A mi familia cuyo amor y paciencia me motivan a seguir adelante.
- A mis amigos Iván y Felipe cuya guía y compañerismo fueron parte esencial para alcanzar mis metas.
- Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por la Beca No. 372756 para mis estudios de especialidad
- Al ITESO por proporcionar las instalaciones, equipos y laboratorios, además del seguimiento y apoyo de los profesores del programa;
- Finalmente, pero no menos importante, a mis asesores Héctor y Luis por el apoyo técnico en la resolución del problema de mi trabajo de grado, así como en la revisión de este documento.

RESUMEN

Este trabajo aborda el tema de los convertidores corriente directa/corriente directa (**DC/DC**), cuyo desarrollo se presenta como un sistema embebido de aplicación automotriz, que incluye las explicaciones del circuito electrónico, el programa del sistema, la ley de control del convertidor y el proceso que asegura la calidad del sistema. El convertidor DC/DC atiende a la problemática de elevar el voltaje suministrado por una batería de plomo de 12V a un voltaje de corriente continua de hasta 400 V con una potencia máxima de 200W y eficiencia superior al 90%. Típicamente los convertidores de voltaje tienen una ganancia que oscila entre 10 y 15 unidades, por lo que una ganancia superior a las 30 unidades se considera alta ganancia. Para cubrir el requisito de alta ganancia el presente proyecto propone la utilización de un convertidor cuya topología se considera novedosa respecto a circuitos tradicionales como el BOOST o el Flyback. Lo novedoso del funcionamiento del convertidor DC/DC radica en su configuración capaz de cargar dos capacitores en paralelo y descargarlos en serie a la vez que combina la topología tradicional del convertidor Flyback. Adicionalmente, se incorpora como elemento de conmutación un MOSFET de SiC, cuya capacidad de operación en temperaturas superiores a 150 ° C lo convierte en un elemento ideal para aplicaciones automotrices.

Los convertidores DC/DC como el que se presenta en este documento pueden tener aplicaciones automotrices como cargadores de baterías de alto voltaje, convertidor alimentador para inversores de corriente, convertidor alimentador para motores de DC. Adicionalmente pueden existir aplicaciones no automotrices como convertidor alimentador para inversores donde las fuentes de energía son de bajo voltaje como es el caso de los paneles solares.

El convertidor DC/DC que aquí se trata tiene la desventaja que el voltaje entregado a la salida es altamente sensible a las variaciones del voltaje de la batería que se toma como entrada al sistema. Esto se explica rápidamente si se considera que el voltaje de salida es directamente proporcional a la ganancia. Para contrarrestar este efecto indeseable se ha incluido la propuesta de una ley de control que permita robustecer el sistema. La ley de control que se presenta se basa en la teoría de controladores por modos deslizantes, ampliamente conocida en la disciplina del control automático. Claro que su uso no se justifica por su popularidad, sino por las ventajas que ofrece esta técnica de control a sistemas conmutados. Brevemente se puede afirmar que los modos deslizantes son una excelente herramienta de análisis aplicable en sistemas cuya variable de control es la conmutación encendido-apagado.

Tomando en cuenta lo mencionado en los párrafos anteriores, es claro que el circuito convertidor DC/DC requiere de elementos externos que hagan la función del módulo de control. Para ello se ha incorporado al sistema un microcontrolador encargado de las tareas de cálculo que requiere la ley de control, así como la conversión de las señales analógicas a digitales, la seguridad del sistema y la comunicación con el usuario, o bien, otros módulos en una red automotriz. Ya incorporados todos los componentes físicos del sistema se abordará el desarrollo del programa que ejecutará el microcontrolador, llegando así a la integración de todos los elementos requeridos por un sistema de categoría embebido.

Para la implementación del programa del sistema se han seguido varias de las recomendaciones relacionadas con la construcción por capas de la especificación de AUTOSAR. Estas incluyen la capa del microcontrolador, la de abstracción de software y la de servicios. En la capa de servicios, se incluye el desarrollo del sistema operativo con capacidades de interrupción preferente a tareas de mayor prioridad. Esta capacidad del sistema para cubrir requisitos de tiempo de ejecución invariablemente estrictos y periódicos es de la más alta importancia para permitir que la ley de control funcione correctamente.

Aunque el desarrollo del proyecto se presenta como una secuencia de actividades a seguir, es importante tener en mente que todo proyecto debe seguir una metodología de desarrollo, es por ello que la documentación se presenta en base al Ciclo-V de desarrollo de sistemas. Como primera fase se describen los requisitos del sistema, seguido por el diseño, la implementación y por último las pruebas. Este patrón se presta a varias interpretaciones, sin embargo lo que se debe tener presente es que en cada etapa de la implementación se realizan las pruebas correspondientes, sin embargo como primeras fases del desarrollo se dedica tiempo de exclusividad para el análisis de requisitos y el diseño del sistema basado en dichos requisitos. Es así, que siguiendo una metodología de desarrollo, se busca obtener un producto final de calidad.

Finalmente, es importante subrayar que todos los elementos que componen el presente proyecto cubren ampliamente la definición de un sistema embebido, que si bien suele considerarse a este tipo de sistemas como aquellos que contienen los elementos requeridos para cubrir una funcionalidad específica, dicho conjunto de elementos suele limitarse a los componentes físicos y los programas que en ellos se ejecutan. Sin embargo, en el proyecto que aquí se presenta también se considera embebido aquello intangible que escapa a la visión de los usuarios, pero que debe ser importante para los que con pasión participen de esta área de la ingeniería, me refiero al análisis, el diseño, los procesos y la calidad.

ABSTRACT

The energy draw on is a topic that has attracted a large number of researchers, entrepreneurs, and in general, a significant sector of society. The causes are diverse, whether due to concern about climate change, economics, knowledge or the limitations inherent in the availability of energy resources provided by the earth versus the energy demand of a growing world population. This document focuses in direct current / direct current (DC / DC) converter subject, which in turn is developed as an embedded system for automotive applications. The document includes the explanations of the electronic circuit, the system program, the control law of the converter and the process that ensures the quality of the system.

The DC / DC converter addresses the problem of raising the voltage supplied by a 12V lead-acid battery to a DC voltage up to 400V with a maximum power of 200W and a 90% efficiency at least. Typically the voltage converters have a gain range from 10 to 15 units, so a gain of more than 30 units is considered high gain. To cover the high gain requirement this project proposes to use a topology considered novel regarding traditional circuits like BOOST or Flyback. The DC / DC converter novelty lies in its configuration which is able to load two capacitors in parallel and discharge them in series while combining the traditional topology of the Flyback converter. In addition, a SiC MOSFET is incorporated as a switching element, whose operating capacity in temperatures above 150 ° C makes it an ideal element for automotive applications.

DC / DC converters such as the one of this document may have automotive applications such as high voltage battery chargers, feeder converter for inverters, and power inverter for electrical motors. In addition there may be non-automotive applications as a feeder converter for inverters where the power sources are low voltage range as instance solar panels devices.

The DC / DC converter discussed here has the disadvantage that the voltage delivered to the output is highly sensitive to battery voltage variations which in turn supplies the input to the system. This is explained because the output voltage is directly proportional to the gain. To counteract this undesirable effect, It has been included a law of control proposal in order to strengthen the system. The control law presented is based on sliding modes controller theory, widely known in automatic control discipline. Of course, its use is not justified by its popularity, but by the advantages offered by this control technique when applied to switched systems. Briefly, it can be said that sliding modes are an excellent analysis tool applicable in systems whose control variable is on-off switching.

Taking into account what is mentioned in the previous paragraphs, it is clear that the DC / DC converter circuit requires external elements in order to implement the control functionality. For this purpose, a microcontroller has been incorporated into the system which in turn is responsible of the calculation tasks required by the control law, as well as the conversion of analogue to digital signals, system security and communication with the user, or other modules in an automotive network. Once all the physical components are incorporated by the system, it will be addressed the program development which runs the microcontroller. The hardware and software integration brings the system closer to the embedded system kind of definition.

Several of the recommendations related to the layered construction of the AUTOSAR specification have been followed for the implementation of the system program. These include the microcontroller layer, the software abstraction layer and the services layer. Services layer includes the operating system development with preferred interruption capabilities to higher priority tasks. The system ability to cover strict and periodic runtime requirements is of the utmost importance in order to allow the control law good working behavior.

Although the project development is presented as a sequence of activities to follow, it is important to keep in mind that every project must follow a development methodology, which is why the documentation is based on V-Cycle methodology for systems development. The first phase describes the system requirements, followed by the design, implementation and finally the tests. This pattern lends itself to several interpretations, however what must be kept in mind is that tests are performed at each implementation stage. However, exclusivity time is dedicated to the analysis of requirements and design. Thus, A development methodology is followed with the sake of obtain a final quality product.

Finally, it is important to underline that all elements which are part of this project largely cover the embedded system definition. This kind of systems are usually considered as those that contain themselves all the elements in order to comply with a specific functionality. However, the set of elements is usually limited to hardware and software. This project also takes in account working activities like analysis, design, adherence to processes and quality as part of the embedded system. All of those should be important development elements for those who passionately participate in this area of engineering.

TABLA DE CONTENIDO

LISTA DE FIGURAS	9
LISTA DE TABLAS	10
1. INTRODUCCION	11
1.1. CONTEXTO DE LOS CONVERTIDORES DE VOLTAJE.....	12
1.2. PROBLEMÁTICA A RESOLVER	12
1.3. OBJETIVOS.....	13
1.4. DESCRIPCIÓN FUNCIONAL	13
1.5. REQUERIMIENTOS DEL SISTEMA	13
1.5.1. BATERÍA.....	14
1.5.2. COMUNICACIONES.....	14
1.5.3. CONTROL	14
1.5.4. SISTEMA OPERATIVO.....	15
1.5.5. OPERACIÓN EN BAJO VOLTAJE.....	15
1.5.6. ESTADO DE FALLA	15
1.5.7. POTENCIA.....	15
2. MARCO TEÓRICO	16
2.1. PRINCIPIOS BÁSICOS DE CIRCUITOS ELECTRÓNICOS DE POTENCIA	18
2.2. CONTROL DE CONVERTIDORES POR CONMUTACIÓN.....	22
3. DISEÑO DEL SISTEMA DE CONTROL EMBEBIDO.....	27
3.1. ARQUITECTURA DE SOFTWARE	28
3.2. SISTEMA OPERATIVO.....	28
3.2.1. TAREAS OCULTAS.....	28
Offsets.....	29
Enmascaramiento mínimo para tareas y longitud de conteo.....	29
3.2.2. TIEMPO BASE PARA EL SO	29
3.2.3. TABLA DE VERDAD PARA EL ALGORITMO	30
3.2.4. ESTRUCTURA DE ARCHIVOS.....	30
3.2.5. DATOS MIEMBRO DE LA ESTRUCTURA DEL TM	31
3.2.6. DISEÑO DE LA ESTRUCTURA DEL BUFFER.....	32
3.2.1. FUNCIONAMIENTO DEL TCB	33
3.2.2. GUARDAR Y RESTAURAR EL CONTEXTO DE LA TAREA	33
3.3. MANEJADOR DE MENSAJES DE CAN	35
3.4. TRANSMISIÓN DE MENSAJES CAN	37
3.5. CIRCUITO CONVERTIDOR	37
3.6. CONTROLADOR DEL CONVERTIDOR.....	39
3.7. IMPLEMENTACIONES.....	41
3.7.1. ALGORITMO DE PLANEACIÓN	41
3.7.2. IMPLEMENTACIÓN DEL ACTIVADOR DE TAREAS.	44
3.7.3. DESCRIPCIÓN DEL DESPACHADOR DE FUNCIONES	44
3.7.4. DESCRIPCIÓN DE LA FUNCIÓN DE TERMINACIÓN.....	45

3.7.5.	DESCRIPCIÓN DE LAS OPERACIONES DE BACKGROUND	46
3.7.6.	SERVICIO DE INTERRUPCIÓN DEL SO.....	47
	Cambio de contexto.....	47
	Restauración de contexto	47
	Rutina de interrupción del SO.....	48
	Ejemplo para la declaración de una tarea	48
	Ejemplo de activación de una tarea que active otra tarea	48
3.7.7.	OPERACIÓN	49
3.7.8.	CONFIGURACIÓN DEL MANEJADOR (DRIVER) DE CAN.....	50
	Fuente del Reloj y Velocidad de Transmisión:.....	50
3.7.9.	CIRCUITO CONVERTIDOR	50
3.7.10.	SISTEMA DE CONTROL.....	51
4.	RESULTADOS DE LA IMPLEMENTACIÓN EMBEBIDA.....	55
4.1.	CASOS DE PRUEBA DEL PLANIFICADOR	56
4.1.1.	BPS_TC_1	56
4.1.2.	BPS_TC_2.....	59
4.1.3.	BPS_TC_3.....	60
4.2.	CASOS DE PRUEBA DEL DESPACHADOR.....	61
4.2.1.	TMNP_TC_1	62
4.2.2.	TMNP_TC_2.....	63
4.2.3.	TMNP_TC_3.....	64
4.2.4.	TMNP_TC_4.....	65
4.3.	CASOS DE PRUEBA CAMBIO DE CONTEXTO.....	67
4.3.1.	PIV_TC1	67
4.3.2.	PIV_TC_2	69
4.3.3.	PIV_TC_3	70
4.3.4.	PIV_TC_4	72
4.3.5.	PIV_TC_5	74
5.	CONCLUSIONES.....	76
	REFERENCIAS	78

LISTA DE FIGURAS

Figura 1 Principio de operación de los convertidores de corriente directa.	19
Figura 2 Fases de operación del convertidor boost y su circuito electrónico equivalente.	19
Figura 3. Convertidor con topología flyback	20
Figura 4. Circuito convertidor de elevación con inductor acoplado	21
Figura 5. Circuito en lazo cerrado de un convertidor controlado por histéresis	23
Figura 6. Circuitos de control de voltaje por PWM.....	24
Figura 7. Arquitectura por capas de un programa embebido de conformidad con AUTOSAR.....	28
Figura 8. Estructura de archivos del sistema operativo.....	31
Figura 9. Miembros de la estructura de configuración de las tareas.	31
Figura 10. Arreglo de la estructura de configuración de las tareas.	32
Figura 11. Miembros de la estructura del bloque de control de tareas.....	32
Figura 12. Rutina de activación de tareas	33
Figura 13. Diagrama a bloques de la rutina para ejecutar tareas.....	34
Figura 14. Secuencia de ejecución durante un cambio de contexto.....	35
Figura 15. Rutina de operación del manejador de mensajes de CAN.....	36
Figura 16. Diagrama de transmisión de mensajes por CAN	37
Figura 17. Topología del convertidor de elevación de alta ganancia.....	37
Figura 18. Interfaz electrónica del convertidor con el sistema de control.	40
Figura 19. Implementación del algoritmo de planeación.....	42
Figura 20. Diagrama a bloques del algoritmo de planeación de tareas.....	43
Figura 21. Diagrama de secuencia del activador de tareas.	44
Figura 22. Diagrama de secuencia del despachador de tareas.	45
Figura 23. Diagrama de bloques de la función terminar.....	46
Figura 24. Secuencia de llamadas de función de la rutina de background.....	46
Figura 25. Diagrama a bloques de la rutina de background.....	47
Figura 26. Diagrama de tiempo que muestra la ejecución de las tareas y el tiempo base.....	49
Figura 27. Fuente de reloj para el módulo de CAN.	50
Figura 28. Plantilla PCB del circuito convertidor.	51
Figura 29. Diagrama del modelo del sistema en su representación a bloques.	51
Figura 30. Representación en diagrama a bloques de la ecuación 2.6.....	52
Figura 31. Representación en diagrama a bloques del modelo simplificado de operación del diodo.....	52
Figura 32. Respuesta del sistema a una entrada tipo escalón.....	53
Figura 33. Respuesta lazo cerrado utilizando un control SM con retroalimentación de corriente.....	54

LISTA DE TABLAS

Tabla 1. Correlaciones de Calidad (House of Quality).....	14
Tabla 2. Periodicidad de las tareas.....	29
Tabla 3. Máscaras de las tareas para el scheduler.....	29
Tabla 4. Algoritmo de distribución de las tareas.....	30
Tabla 5. Tabla de terminales para medir la carga de una tarea.	43
Tabla 6. Lista de materiales que componen el circuito convertidor.....	50

1. INTRODUCCION

Resumen: En este capítulo se presenta de manera general el trabajo desarrollado, así como sus principales objetivos.

1.1. Contexto de los Convertidores de Voltaje

En años recientes las fuentes de energía renovable han captado la atención de investigadores y emprendedores, cuyo objetivo en común es encontrar tecnologías que permitan aprovechar los recursos energéticos disponibles, de forma eficiente, limpia y sustentable. La electricidad se ha consolidado como la forma de energía mejor adaptable a algunos de los requerimientos del hombre, tales como: disponibilidad, almacenamiento, limpieza, bajo riesgo y bajo costo. Todos estos factores han propiciado el desarrollo de tecnologías de conversión de voltaje en diversos sectores, siendo el automotriz el de principal interés para el proyecto que aquí se describe.

Para la implementación del presente proyecto se hizo uso del modelo de desarrollo ciclo-V (o también llamado “V-cycle”). Este ciclo consiste en cuatro segmentos que se identifican por los procesos que los describen, estos son requerimientos, diseño, implementación y pruebas. Es mucho más común encontrar este ciclo de desarrollo en proyectos de software. Considerando que la funcionalidad del hardware depende del software, no es extraño que sea conveniente aplicar esta técnica de trabajo, más aún, debido a ello la selección del hardware se adapta naturalmente al ciclo-V. Aunque lo razonable sería comenzar este documento con la descripción de requerimientos, se ha incluido un capítulo intermedio con el marco teórico de los principios fundamentales que se utilizan en el resto del contenido. Los principales elementos de esta sección son teorías de convertidores de elevación de voltaje y métodos de control para los mismos.

1.2. Problemática a Resolver

Existen diferentes campos de aplicación automotriz para los circuitos electrónicos de potencia. Uno de ellos, es el de los convertidores de corriente directa a corriente directa (DC/DC). El interés del presente trabajo se orienta a los convertidores de alta ganancia para elevar el voltaje. Dicho objetivo surge de la necesidad de utilizar baterías de bajo voltaje, que ofrecen mayor seguridad al usuario, y motores de tracción de alto voltaje que permiten tener un mejor aprovechamiento de la energía durante la conducción de un vehículo eléctrico o bien uno híbrido. Existen varias topologías de convertidores, por ejemplo, en teoría, el convertidor tipo “boost” el cual puede proveer de altos valores de ganancia con el inconveniente de tener un alto porcentaje de activación durante la conmutación del interruptor (1). Sin embargo, en la práctica la ganancia por elevación de voltaje está limitada por el efecto de la suma de las resistencias del conmutador, diodo rectificador, capacitor y el inductor, además de otros problemas como la interferencia electromagnética (2). Para resolver estos problemas se ha seleccionado una topología novedosa (3), que permite alcanzar altos niveles de ganancia para la elevación de voltaje utilizando un reducido número de elementos y con alto rendimiento. La selección de los elementos que componen el convertidor a utilizar se describe como parte de las instancias de interés que se desarrollan a través del ciclo-V.

El convertidor a utilizar cuenta con características apreciadas, tales como robustez a cambios en la carga, sin embargo, el voltaje de salida es significativamente sensible a variaciones de potencial a la entrada. Estas variaciones de voltaje suelen ser comunes en cualquier aplicación eléctrica, en el caso del segmento automotriz el problema se agrava. Para contrarrestar las perturbaciones derivadas de variaciones de voltaje en la fuente de alimentación se propone un método de control que regule la

salida para alcanzar un potencial constante. El diseño del controlador se describe en el segmento del ciclo-V enfocado al diseño, sin embargo, también se incluyen secciones dedicadas a los requerimientos del controlador, así como su implementación y prueba en los capítulos respectivos.

Finalmente, a lo largo de los segmentos del ciclo-V se incluyen los módulos de software utilizados para obtener un convertidor funcional, seguro, y capaz de integrarse con otras aplicaciones a través de una red de comunicaciones CAN. Los módulos que componen el software son drivers, capas de abstracción de hardware, bibliotecas de funciones matemáticas, manejo de memoria, sistema operativo y aplicación. Todos los elementos reunidos en este proyecto y descritos a lo largo del presente documento integran un sistema embebido capaz de recordarnos lo complejo y fascinante que puede ser la capacidad creativa del hombre convertida en tecnología.

1.3. Objetivos

Construir un prototipo de convertidor DC/DC de alta elevación de voltaje que utilice como alimentación de entrada al circuito una batería Acido-Plomo de 75 Amperes a 12 Volts como las utilizadas comúnmente en aplicaciones automotrices, y que entregue a la salida del convertidor un voltaje de 400 Volts. La topología del convertidor a utilizar fue desarrollada por Juu Liang et. al. (3), que presenta una configuración “*flyback*” modificada que permite alcanzar alta ganancia en la relación de elevación de voltaje con eficiencia superior al 90%, y potencia de hasta 1000 W.

El Convertidor seleccionado, es altamente sensible a variaciones de voltaje de alimentación, por lo que para alcanzar el objetivo es necesario desarrollar un sistema de control que estabilice el voltaje de salida en el valor deseado. Para ello es necesaria la construcción de un sistema embebido que implemente la ley de control para el convertidor que se caracteriza por ser un sistema no-lineal de estructura variable.

1.4. Descripción Funcional

El Convertidor de voltaje realizara su función de elevación tomando una entrada no regulada de 12V. A la salida del convertidor se obtendrá un voltaje de salida de 400V regulados. Para el control de la elevación se utiliza un conmutador tipo MOSFET de SiC. La frecuencia de conmutación es fija de 20 kHz. El control se llevará a cabo por un micro-controlador de la familia 9S12X de Freescale.

El sistema embebido se encargará de las funciones de conversión de señales analógicas a digitales, comunicación con la ECU principal la cual ordenará la activación del convertidor. El convertidor reportara su estado y el valor del voltaje a la salida. En caso de falla de comunicación el convertidor no deberá transferir energía de su entrada a la salida.

1.5. Requerimientos del Sistema

Para el planteamiento de requerimientos se ha recurrido al uso de la técnica del “House of Quality”, la cual se basa en asignar propiedades de interés y asociarlas a variables del dispositivo a construir de tal forma que al asignar niveles de importancia se identifican las correlaciones que existen entre las diferentes variables del sistema y así se puede obtener un índice de importancia que permite enfocar el esfuerzo de diseño en los atributos de mayor valor. El resultado para el convertidor se presenta a continuación.

Customer Requirement	Importance	%Importance	↑	↑	↓	↓	↓	↓	↓	↑	↑
			Weight	Hours per Maintenance	Voltage Ripple	Voltage Output	Power	Load	Battery Variations	Feedback Control	CAN Communications
Reliable	5	17%		○	○	○	○	○	○	○	○
Flexible	2	7%	○	△	○	○	○	○	△	△	○
Thermal Stability	3	10%		△	○	○	○	○			
Efficient	4	13%			△	○	○	○	△	△	
Accurate	4	13%	△	○	○	○	○	○	○	○	△
Robustness	4	13%		○	○	○	○	○	○	○	△
Magnetic Compatibility	4	13%	△		○	△	○	△	△	△	△
Life	4	13%		○	○	○	△	○	○	○	
			0.47	3.47	5.13	6.13	3.53	4.73	4.63	4.63	1.10
			1.43%	10.59%	15.68%	18.74%	10.79%	14.46%	14.15%	14.15%	3.36%

Tabla 1. Correlaciones de Calidad (House of Quality)

A partir del planteamiento del HoQ1 se plantean los requerimientos para el convertidor, los cuales están agrupados como se muestra a continuación.

1.5.1. Batería

- El voltaje de la batería **deberá** suministrar 12 voltios con una tolerancia de +/- 30%.
- El suministro de energía **deberá** ser proporcionado por una batería con tecnología ácido-plomo.
- La corriente nominal que **deberá** proporcionar la fuente es de 20 Amperios mínimamente.

1.5.2. Comunicaciones

- Los parámetros del sistema convertidor **deberán** poder ser programados a través de la interfaz del Unidad de Control Electrónica(ECU)
- El sistema convertidor **deberá** ser capaz de establecer comunicación a través de una red con protocolo CAN.
- El sistema convertidor **deberá** encender sólo bajo petición de un ECU maestro conectado a la red CAN.

1.5.3. Control

- El sistema convertidor **deberá** mantener su valor de referencia a variaciones en la carga de 10 ohms/seg.
- El sistema convertidor **deberá** incorporar un sistema de control en lazo cerrado.
- El sistema convertidor **deberá** incorporar algoritmos de cálculo aritmético de punto fijo de 16 bits.
- El suministro de energía principal **deberá** incluir un interruptor manual.
- La batería **deberá** estar protegida contra sobrecargas con un fusible térmico.
- El voltaje de rizo **deberá** ser de +/- 2% respecto al valor de voltaje de salida.

- El voltaje de salida **deberá** ser de 400V con +/- 5V de tolerancia.

1.5.4. Sistema Operativo

- El sistema operativo **deberá** implementar funciones de interfaz (API), que guarden el estado actual de todas las interrupciones, deshabilitar todas las interrupciones que estén habilitadas e identificar el comienzo de una sección crítica.
- El sistema operativo **deberá** incluir un despachador de tareas que será llamado para realizar modificar los siguientes estados de cualquier tarea:
 1. – Activar Tarea.
 2. – Terminar Tarea.
- El sistema operativo **deberá** Guardar el Contexto de la tarea interrumpida y Restaurar el contexto a través del uso de API's que realicen al cambio de contexto.
- Las funciones de cambio de contexto **deberán** proveer como argumento de la función un apuntador a la pila (*stack*) de la tarea interrumpida.
- La función de Guardar contexto **deberá** llamarse inmediatamente después de una interrupción tipo CAT 2.
- El sistema operativo **deberá** soportar interrupciones periódicas y por evento.
- El concepto de máscara **deberá** ser utilizado para cambiar el estado de una tarea a "Listo(Ready)":
 1. Cuando el contador binario y la máscara sean idénticos, la tarea será ejecutada.
 2. La máscara deberá ser un numero definido por la ecuación: $Máscara = (2^n) - 1$.

1.5.5. Operación en bajo voltaje

- El sistema convertidor **deberá** identificar una condición de bajo voltaje.
- El sistema convertidor **deberá** tener un indicador que muestre la condición de bajo voltaje.
- La señal de bajo voltaje será la única que **deberá** mostrarse durante el tiempo de operación en bajo voltaje.

1.5.6. Estado de falla

- El sistema convertidor **deberá** apagarse después de 3 segundos de haberse interrumpido la comunicación.
- El sistema convertidor **deberá** apagarse después de 10 segundos en que voltaje promedio no corresponda con el voltaje de operación especificado.

1.5.7. Potencia

- Los circuitos de la tarjeta de control principal deberán estar aislados de los circuitos de potencia.
- El sistema convertidor deberá suministrar una potencia máxima de 200 W.
- El sistema convertidor deberá operar en entornos con temperaturas máximas de 50°C.
- El sistema convertidor deberá tener una eficiencia energética mínima de 90%.
- El sistema convertidor deberá ser cumplir con las directrices RoHS.
- El sistema convertidor deberá operar por al menos 3000 horas sin mantenimiento.

El sistema convertidor deberá mantenerse en operación con variaciones de voltaje de 100 V/seg.

2. MARCO TEÓRICO

Resumen: En este capítulo se presenta los antecedentes del problema a resolver, así como su contexto

Probablemente las fuentes de energía alternativa más ampliamente utilizadas en la época actual son las construidas a través de celdas de carga. Una celda de carga se basa en un proceso electroquímico: hidrógeno y oxígeno reaccionando para generar energía eléctrica. Las celdas de carga son utilizadas como la fuente principal en redes de suministro, vehículos o en aplicaciones portátiles. Por ejemplo, en 2008 la empresa constructora de aviones Boeing mostró que se podía volar un aeroplano tripulado energizado con celdas de hidrógeno.

Como el voltaje de las celdas de carga suele ser muy pequeño, así como la carga variable (entre 0.4 V a carga plena y 0.8v sin carga), las celdas deben conectarse en serie para obtener un voltaje útil. Por ejemplo, 250 celdas conforman una batería de 100V a plena carga, por esta razón, los circuitos de electrónica de potencia deben ser capaces de generar elevaciones de voltaje capaces de mantener voltajes estables ante variaciones en la fuente y/o la carga. Las dificultades para construir tales tipos de circuitos electrónicos se incrementan ante la necesidad de mantener la magnitud de rizo en la corriente de entrada en un nivel bajo. En el caso de ser una aplicación automotriz, se debe considerar que, ante una falla en el sistema de baterías, el conductor debería tener la posibilidad de continuar su trayecto a un nivel de potencia limitado hasta llegar a un lugar seguro. Este tipo de requerimientos implican un reto completamente nuevo en el diseño de sistemas electrónicos de potencia: la necesidad de implementar convertidores DC/DC por módulos para mejorar la confiabilidad del sistema.

Las celdas no pueden responder rápidamente a fluctuaciones en la carga. Un convertidor en serie entre la celda y la carga no es suficiente, porque una fluctuación de la corriente de carga se refleja directamente en una fluctuación en la celda, disminuyendo su vida útil. Una posible solución es utilizar dos convertidores, uno en serie como el que se describió, y uno en paralelo, el cual al estar conectado a una batería podría realizar la compensación ante fluctuaciones.

Tratándose específicamente de vehículos eléctricos, se necesitan baterías que provean suficiente energía durante la aceleración. Las celdas de carga son una opción, sin embargo, existen otros como las de Nickel o Litio, estas últimas comúnmente utilizadas. Este tipo de baterías también se conforma por varias celdas conectadas en serie alcanzado voltaje de 250 volts, sin embargo, su voltaje de operación suele estar en el rango de los 150 a 270 volts. Un arreglo masivo de baterías requiere de sistemas sofisticados de carga para lograr el mejor desempeño posible, la mayor vida útil al mismo tiempo que se protegen las grandes inversiones que requieren este tipo de baterías. Para trenes de potencia de hasta 100kW, el voltaje nominal de operación es de unos 400 V, por lo tanto, durante los periodos de aceleración, el voltaje de la batería tiene que ser elevado cuando es entregado al inversor. Más aún, el circuito electrónico debe asegurar en su salida, el voltaje constante que permita al consumidor despreocuparse por los cambios de voltaje en las terminales de la batería.

Así mismo, como la carga es variable, el circuito de elevación requiere de un circuito de control para lograr las magnitudes de voltaje constante requeridas. Como requerimientos adicionales, para el uso de aplicaciones automotrices, los circuitos electrónicos de potencia necesitan cubrir rasgos tales como bajo costo, minimización de componentes, tamaño, peso reducido, eficiencia de conversión en un amplio rango de potencia, diseño compacto, baja interferencia electromagnética, confiabilidad y seguridad. Debido a lo anterior, concebir un convertidor para aplicaciones automotrices significa todo un reto de investigación y desarrollo en electrónica de potencia y de un sistema embebido para su control.

2.1. Principios Básicos de Circuitos Electrónicos de Potencia

Analizando diferentes aplicaciones de potencia, se puede observar que el propósito de los circuitos electrónicos de potencia es controlar las variables de salida, por ejemplo, voltaje constante en una carga variable o bien una fuente de energía variable. Por lo tanto, sin importar la aplicación, el circuito de potencia se encuentra entre la entrada no regulada y la carga variable. El circuito de potencia puede concebirse como un sistema procesador de potencia. El primer requisito de un circuito de potencia es procesar la energía eficientemente (4). Esto se traduce en que no se deben utilizar elementos resistivos, esto es complicado por las características resistivas intrínsecas de cualquier dispositivo. Considérese por ejemplo que un mili ohm de resistencia podría disipar 10 watts de potencia en una medición de 100 A de corriente. Aunado a lo anterior, también debe considerarse la disipación de calor, que en cualquier caso es el resultado de las pérdidas de energía en el circuito. Esto conduce a la reducción del tiempo de vida del circuito y su confiabilidad.

A continuación, se describe el principio básico de operación de un circuito elevador de DC/DC que considera, como entrada al proceso de elevación un voltaje de entrada V_{in} variable y entregar un voltaje constante V_{out} a su salida sin importar el valor de la carga R . La estructura más simple que se puede utilizar se basa en el uso de un inductor, un capacitor y un medio de conmutación como se muestra en la Figura 1.a. Cuando el conmutador está en la posición 1(fase 1), la energía es transferida desde la batería al inductor L , cargándolo (Figura 1b). La energía es almacenada en su campo magnético. Cuando el conmutador se mueve a la posición 2(fase 2), el inductor transfiere su energía a la carga (Figura 1.c), esto es, el inductor es descargado. La posición del conmutador varía periódicamente, todo el tiempo, en una operación cíclica. La función del capacitor es mantener el voltaje de salida durante la fase. 1. Como el objetivo es tener un voltaje constante a la salida la capacitancia debe tener un valor grande (unos cientos de μF), y la duración de la fase debe ser muy corta para evitar una descarga total del capacitor. Esto implica que la conmutación debe ocurrir con alta frecuencia. Regularmente la frecuencia está por encima de los 10 Khz. En la segunda fase, la energía del inductor recarga el capacitor C . Como regla básica en teoría de circuitos se debe recordar que al momento de la transición la corriente del inductor no puede cambiar de dirección, lo que se traduce en que la corriente del inductor carga el capacitor C con el voltaje de polaridad opuesto a V_{in} . Por lo tanto, la polaridad de V_{out} será opuesta a V_{in} en el convertidor mostrado en la Figura 1.

Para asegurar un voltaje V_{out} constante ante variaciones de V_{in} , se puede cambiar el tiempo de carga del inductor. Por ejemplo, si el voltaje V_{in} disminuye al aumentar el tiempo de conexión de la fase 1, se transferiría la misma cantidad de energía que con la batería cargada inicialmente, por lo tanto, en la fase 2 el voltaje en V_{out} permanecerá igual. Una operación similar se puede realizar si hay cambios en la carga. Resulta evidente que la acción de conmutación es esencial para mantener un voltaje deseado a la salida.

Para la implementación electrónica del conmutador, se puede utilizar un transistor T y un diodo D como se muestra en la Figura 1.d. La acción de conducción del transistor se denota por la función $d(t)$. Cuando la señal de activación tiene un valor lógico 1(Figura 1.e), el transistor se activa y el inductor se carga, el diodo será polarizado inversamente bloqueando el flujo de corriente a través de él. Cuando la señal de activación tiene el valor lógico de 0, el transistor está en corte, la corriente del inductor encuentra un camino para fluir a través del diodo. De esta forma el diodo actúa como un conmutador automático

síncrono. Para mantener un voltaje constante V_{out} sin importar las variaciones en la batería y la carga, se puede variar el porcentaje de tiempo de encendido manteniendo constante el periodo T_s .

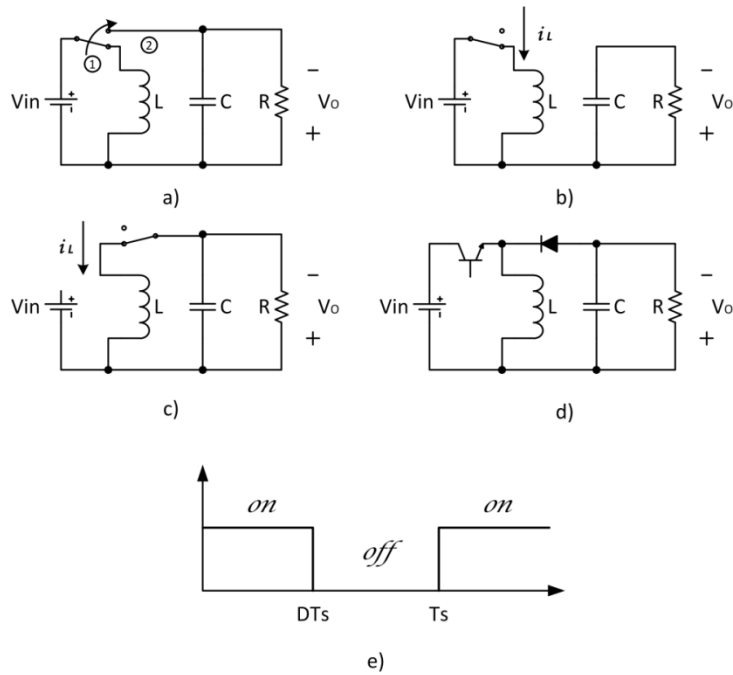


Figura 1 Principio de operación de los convertidores de corriente directa.

Existen diferentes estructuras o topologías de circuitos de elevación. En la Figura 2 se muestra un elevador conocido como “*boost-converter*”. Su principio de operación se puede explicar de la misma forma como se describió el circuito de la Figura 1, con la diferencia que cuando el transistor funciona como interruptor abierto la corriente fluye a través del diodo, descargando la energía del inductor y la batería sobre el capacitor y la resistencia. Debido a que la batería y el inductor se encuentran en serie durante la descarga, el voltaje de salida V_{out} es mayor a V_{in} .

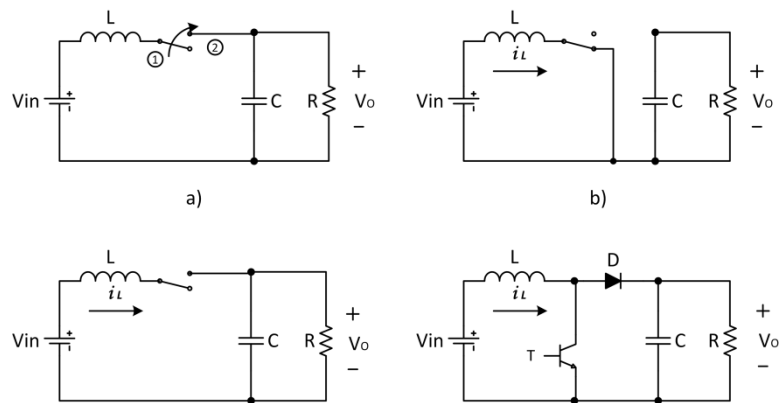


Figura 2 Fases de operación del convertidor boost y su circuito electrónico equivalente.

Otra topología de convertidor elevador de voltaje es la conocida como “*flyback-converter*”, esta se muestra en la figura 3.

Para describir la operación del convertidor “*flyback*”, considere la Figura 4. Cuando el conmutador es activado (Figura 4.a), la polaridad

del voltaje a través del devanado primario es el mismo que el de la batería. Eso provoca que la polaridad el voltaje a través del devanado secundario sea como se muestra en la figura, con los positivos en los puntos del transformador. Como resultado, el diodo D es polarizado inversamente y en estado de apagado. La corriente en el secundario es cero,

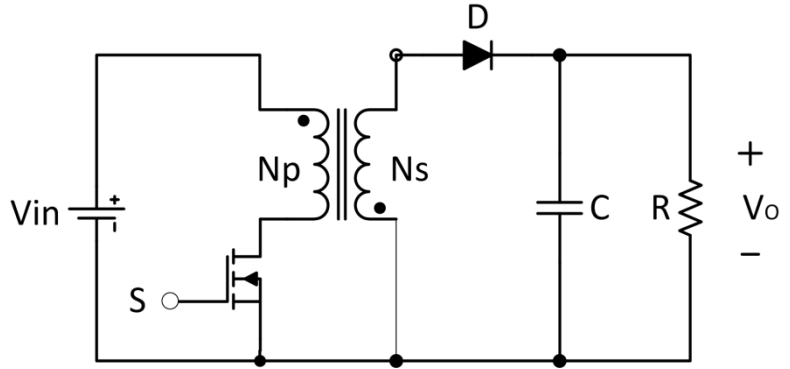


Figura 3. Convertidor con topología flyback

lo que significa que la corriente en el primario también debe ser cero. La inductancia magnética del transformador es cargada con la corriente de entrada, $I_{in}=I_m$. Por lo tanto, la corriente de magnetización en un convertidor tipo “*flyback*” tiene un significativo componente de corriente directa.

De:

$$V_{in} = v_m(t) = L_m \frac{di_m}{dt}, \quad 0 \leq t < DT_s \quad (2.1)$$

se obtiene la expresión para la corriente de magnetización.

$$i_m(t) = i_m(0) + \frac{V_{in}}{L_m} t, \quad 0 \leq t < DT_s \quad (2.2)$$

La corriente de magnetización alcanza su valor máximo en el tiempo DT_s . Los voltajes a través de los devanados, para las polaridades mostradas en la figura, son:

$$v_{pr}(t) = V_{in}; \quad v_{sec}(t) = \frac{N_s}{N_p} v_{pr}(t) = \frac{N_s}{N_p} v_{in} \quad (2.3)$$

El conmutador debe conducir la corriente de entrada, igual a la corriente de magnetización. El voltaje a través del diodo, con las referencias como se muestra en la figura, están dadas por:

$$v_D = -[v_{sec}(t) + V_{out}] = -\left[\frac{N_s}{N_p} V_{in} + V_{out} \right] \quad (2.4)$$

El capacitor tiene que asegurar el voltaje en la carga, que es similar a la fase de encendido de una topología tipo “boost”.

Cuando el conmutador es apagado (Figura 4.b), el flujo de corriente termina. Sin embargo, la corriente de magnetización continúa su flujo de corriente. Como resultado $I_{pr}=I_m$, provocando un flujo de corriente en el secundario con el sentido que se muestra en la figura. La corriente de magnetización polariza el devanado primario con polaridad negativa en la marca del primario, provocando que el devanado secundario sea polarizado negativamente en el nodo con la marca del punto.

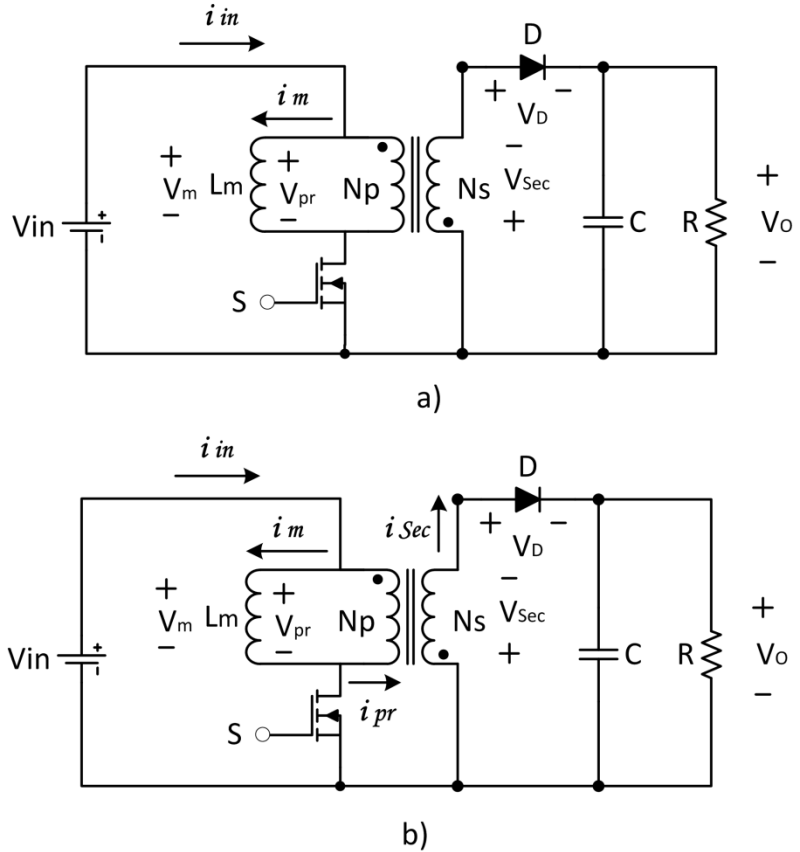


Figura 4. Circuito convertidor de elevación con inductor acoplado

Como consecuencia, el diodo D se polariza directamente permitiendo el flujo de corriente en el secundario. Manteniendo la definición de polaridades de la primera fase de la conmutación, se tiene:

$$v_{sec}(t) = -V_{out} ; \quad V_{pr}(t) = \frac{N_p}{N_s} v_{sec}(t) = -\frac{N_p}{N_s} V_{out} , \quad DT_s \leq t < T_s \quad (2.5)$$

La inductancia de magnetización se descarga con el comportamiento de la ecuación:

$$i_m(t) = i_{m,max} - \frac{1}{L_m} \frac{N_p}{N_s} V_{out}(t - DT_s) ; \quad DT_s \leq t < T_s \quad (2.6)$$

De acuerdo con las orientaciones de la corriente mostradas en la Figura 4.b, esta es la corriente en el primario. El diodo conduce la corriente transportada en el secundario como:

$$i_D(t) = i_{sec}(t) = \frac{N_p}{N_s} \left[i_{m,max} - \frac{1}{L_m} \frac{N_p}{N_s} V_{out}(t - DT_s) \right] \quad (2.7)$$

Se puede concluir que el comportamiento del convertidor con topología de “flyback” se comporta como el convertidor tipo “boost”, con aislamiento debido al dispositivo transformador que propiamente dicho actúa como inductor acoplado.

Un inductor acoplado requiere de un núcleo ranurado para el almacenamiento de la energía. Un material magnético ideal no almacena energía. Por lo tanto, una ranura de aire es un medio no magnético capaz de almacenar energía en serie con la alta permeabilidad magnética del núcleo.

También debe observarse que la corriente de magnetización fluye en una sola dirección, esto es, es positiva durante todo el ciclo de conmutación. Llevando un alto componente de corriente directa. El inconveniente de utilizar este tipo de topología es que si se desea aumentar la capacidad de almacenamiento de energía es necesario aumentar el tamaño de la ranura de aire.

Finalmente, el voltaje de salida del convertidor tipo “*flyback*” es:

$$V_{out} = \frac{D}{1-D} \frac{N_s}{N_p} V_{in} \quad (2.8)$$

Confirmando la semejanza de este tipo de convertidor con el “*boost-converter*”, pero con el incremento en la ganancia debida al inductor acoplado.

2.2. Control de Convertidores por Conmutación

La mayoría de los convertidores DC/DC son diseñados con un sistema de control en lazo cerrado para alcanzar un voltaje regulado a la salida. El objetivo principal es asegurar que el convertidor funcionará con un error pequeño a la salida en estado estacionario, rápida respuesta a cambios en el sistema, bajo sobre impulso, y baja susceptibilidad al ruido, al mismo tiempo que tiene una alta eficiencia y baja emisión de ruido. Esto se logra seleccionando el criterio adecuado de diseño, parámetros del circuito y estrategia de control.

Las estrategias de control más comunes utilizadas en convertidores DC/DC son:

- Controladores por histéresis.

Antes de los circuitos de modulación por ancho de pulso (PWM por sus siglas en ingles), la técnica de control más utilizada fue el control por conmutación con histéresis. Hay dos tipos de controladores por histéresis, los de voltaje y los de corriente. Probablemente el método más simple de regulación de un convertidor DC/DC, es el de histéresis de voltaje como se muestra en la Figura 5., que se compone únicamente de un conmutador con histéresis, el cual compara la salida de voltaje actual con la salida de voltaje deseada. Si la salida de voltaje está muy por debajo del valor deseado el conmutador S_w se activa; y si la salida de voltaje está muy por encima del valor deseado, el conmutador S_w se apaga. Sin embargo, esta solución solo es efectiva para los conmutadores con topología tipo “*buck-converter*”, donde no hay retardo en la fase de la transferencia de energía. Por otra parte, para el tipo de convertidor “*boost*” y “*flyback*”, resulta una técnica inadecuada.

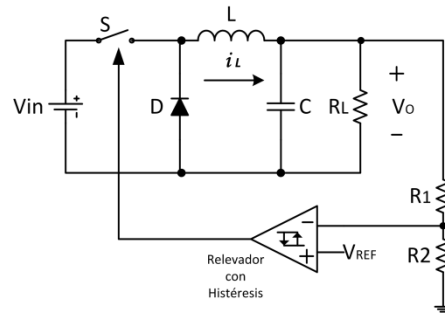
Debe tenerse presente que este tipo de convertidores se caracterizan por tener una función de transferencia de fase no mínima, lo que se traduce en un retardo entre la acción de control y la respuesta en el voltaje de salida. Para solventar este problema se utilizan los controles histéresis por corriente como se muestra en la Figura 5.b. En este método la corriente del inductor es comparada con la compensación del error de voltaje. Si la corriente del inductor es demasiado baja, el conmutador S_w se activa; si la corriente del inductor es demasiado alta, el conmutador S_w se apaga. Como resultado, el control de la corriente permite regular el voltaje de salida.

Sin embargo, los tiempos de histéresis para el encendido y el apagado no se encuentran fijos. La frecuencia de operación es variable y susceptible a los parámetros del convertidor. Por lo tanto, aunque los convertidores controlados por histéresis son simples de implementar, tienen un espectro de ruido impredecible, generando que el ruido por interferencia electromagnética sea más difícil de limitar.

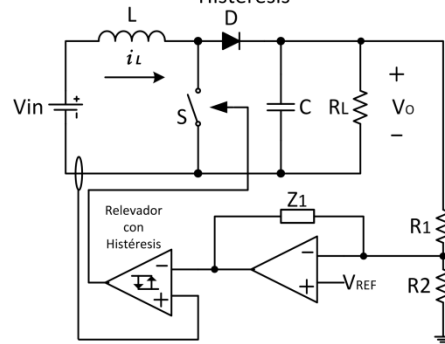
- Controles por modulación de ancho de pulso

Los controles que utilizan PWM a una frecuencia fija son por mucho la técnica de control más popular para la regulación de convertidores DC/DC. Las razones principales son:

1. La disponibilidad de circuitos generadores de PWM de bajo costo.



a) Control por modo de voltaje con Histéresis



b) Control por modo de corriente con Histéresis

Figura 5. Circuito en lazo cerrado de un convertidor controlado por histéresis

2. La necesidad de eliminar las emisiones de los convertidores por conmutación en entornos cada vez más sensibles para los sistemas computacionales y de comunicaciones.

Los convertidores por conmutación son fuentes poderosas de ruido electromagnético. La tarea de contener el ruido se vuelve más simple utilizando frecuencias fijas. Con los filtros adecuados, aterrizajes y blindajes correctos los convertidores por conmutación pueden utilizarse junto con aplicaciones sensibles a la interferencia electromagnética (5).

De manera similar a los controles por histéresis, la frecuencia del controlador por PWM utiliza una frecuencia fija y dos métodos específicos, llamados control por retroalimentación de voltaje y control por corriente programada, o bien modo voltaje y modo corriente respectivamente.

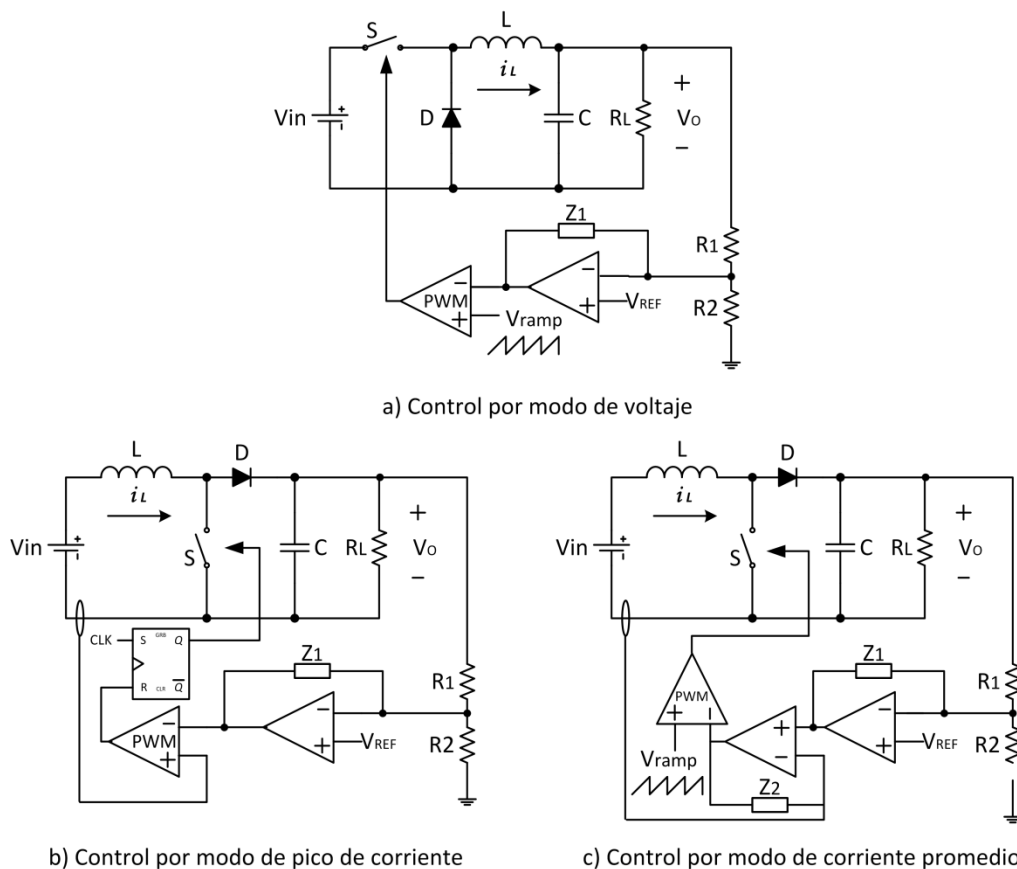


Figura 6. Circuitos de control de voltaje por PWM

El control por modo de voltaje es un control de un solo lazo donde se regula el voltaje de salida cerrando el lazo de retroalimentación entre el voltaje de salida y el porcentaje de encendido (Figura 6.a). El voltaje de salida se compara con una señal de referencia constante V_{ref} , para producir una señal de error, la cual es procesada a través de un compensador Z_f para generar la señal de control. El modulador de ancho de pulso compra esta señal de control con una función tipo rampa generada externamente para obtener la

señal de control deseada para controlar el conmutador de potencia. Típicamente, se necesita un esquema de adelanto de fase para aumentar la inmunidad del voltaje de salida del convertidor contra perturbaciones en el voltaje de entrada. Esto se logra haciendo que el voltaje de pico de la señal rampa sea proporcional a la entrada de voltaje. Dicha implementación por adelanto de fase no es necesaria en el modo de control por corriente.

El modo de corriente es un sistema de control de dos lazos, el cual utiliza un lazo interno de corriente, en conjunto con el lazo de retroalimentación de voltaje. En cierto modo, esto introduce un estado de retroalimentación no lineal, a diferencia del control por modo voltaje que es totalmente lineal. Como se mencionó previamente, este tipo de control se aplica principalmente a los convertidores con topología tipo “*boost*”, que se caracterizan por ser sistemas de fase no mínima. La principal ventaja del modo de control por corriente es que incrementa el margen de estabilidad del sistema y por lo tanto simplifica el diseño del lazo de retroalimentación de voltaje. Existen básicamente dos tipos de controles por modo de corriente, llamados modo de corriente pico y modo de corriente promedio (6).

En el modo de control por corriente pico, el objetivo es forzar la corriente de pico del inductor a seguir una señal de referencia la cual se obtiene del lazo de retroalimentación del voltaje de salida como se muestra en la Figura 6.b. La idea es encender el conmutador S_w periódicamente según la frecuencia fija establecida, y apagarlo cuando el pico de la corriente instantánea alcance el nivel de referencia deseado. La desventaja principal de este método es que es extremadamente susceptible al ruido, dado que la corriente de rampa es utilizada como una señal pequeña en comparación a la señal de referencia. Una segunda desventaja es una propiedad inherente de inestabilidad al exceder el 50 por ciento de activación del periodo de encendido, que conduce a oscilaciones. Típicamente, se lleva a cabo una compensación de pendiente a la entrada del comparador para eliminar esta inestabilidad. La tercera es que la respuesta no es ideal por utilizar el valor de un solo punto de la señal de corriente en vez de valor más general como el promedio.

Para el modo de control por corriente promedio, Z_{f1} el compensador utiliza el voltaje de salida para genera el nivel de corriente programada deseada, que se comparara con la corriente del inductora obteniéndose así el error en la corriente (Figura 6.c). El modulador por ancho de pulso compara el valor del compensador Z_{f2} y la señal de rampa para generar la señal de control deseada.

La principal ventaja de este método sobre el de corriente pico es que se mantiene la estabilidad a pesar de exceder el 50% del ciclo de encendido, provee un buen seguimiento de la corriente promedio y tiene excelentes propiedades de inmunidad al ruido. Sin embargo, debido a que existen dos redes de compensación Z_{f1} y Z_{f2} en serie, el análisis y diseño de optimización no es simple. Este es el principal obstáculo para la utilización de este modo de control.

Otro modo de control, el cual se encuentra en investigación, es el de modos deslizantes. El control por modos deslizantes es un tipo de control no lineal introducido inicialmente como un medio para controlar sistemas de estructura variable. La primera instancia de su aplicación en convertidores DC/DC se reportó en 1983 (7) y 1985 (8). Desde entonces, ha habido un amplio interés en la comunidad científica sobre su desarrollo. La principal ventaja del control por modos deslizantes sobre otros tipos de control no lineal es que el método de implementación es sencillo. Esto hace que sea adecuado para los propósitos de regulación de los convertidores DC-DC. Sin embargo, el problema principal asociado a su aplicación es

su propiedad de frecuencia variable, que provoca pérdidas de potencia excesivas e interferencia electromagnética. Sin embargo, si el problema es manejado correctamente, los modos deslizantes tienen un gran potencial en aplicaciones de todo tipo.

3. DISEÑO DEL SISTEMA DE CONTROL EMBEBIDO

Resumen: En este capítulo se presentan el diseño propuesto como solución al problema de estudio en este trabajo.

En el presente capítulo se muestra el diseño propuesto para cubrir los requerimientos funcionales propuestos en el capítulo anterior. Como cualquier sistema embebido, el diseño se puede segmentar en los elementos de hardware y software, por lo que la descripción del diseño se segmenta en estos dos rubros.

El diseño del control se describe en el segmento de código por ser este el que refleja la implementación de la ley de control, mientras que el modelo utilizado del convertidor se anexa en el segmento dedicado al circuito.

3.1. Arquitectura de Software

La arquitectura del código está basada en las directivas que proporciona AUTOSAR como guía para el desarrollo de software, de tal forma que los módulos desarrollados sean modulares y de fácil mantenimiento. La implementación final se apega a la descripción de capas que se muestra en la Figura 7.

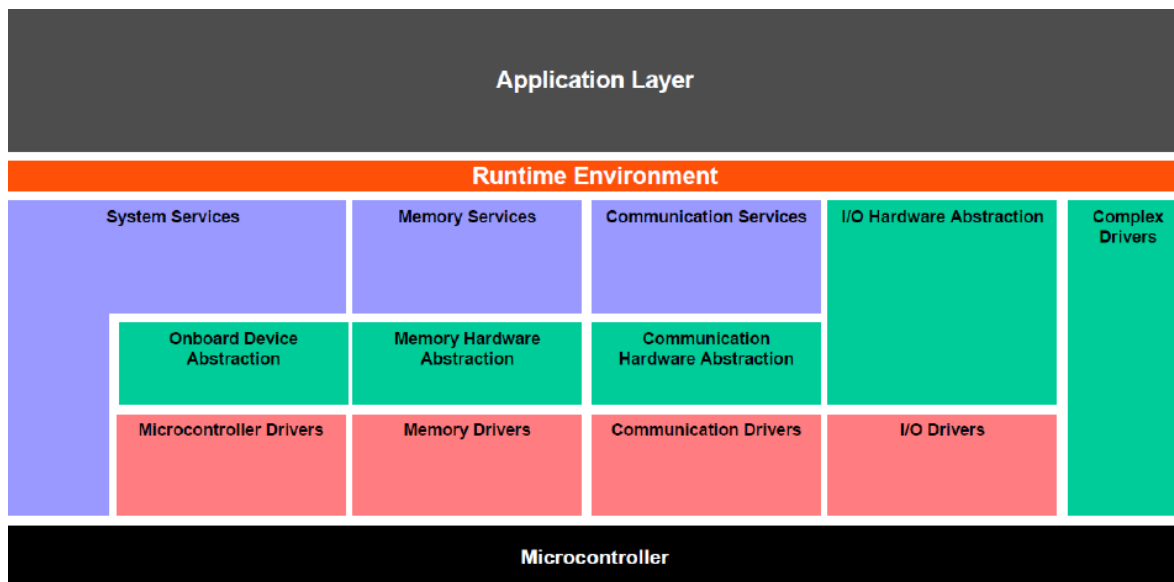


Figura 7. Arquitectura por capas de un programa embebido de conformidad con AUTOSAR.

3.2. Sistema Operativo

En esta sección se describen las consideraciones que dictan el comportamiento de la implementación, así como los diagramas de flujo utilizados para implementar el Sistema Operativo que en general se describe como del tipo “preemptive” de ejecución por prioridad.

3.2.1. Tareas ocultas

La implementación de las tareas se puede relacionar con una progresión binaria, por ejemplo, en el caso de tener siete tareas que se ejecutan periódicamente se plantea la tabla 2, la cual muestra la tarea y su período.

Debido a existen convergencias en el tiempo para una o más tareas, la tarea que se ejecutará primero es la que tenga mayor prioridad, la cual tendrá un retardo en su ejecución igual a cero, asignando retardos u “offsets” a las tareas restantes.

No. de Tarea	Périodo
Task1	1ms
Task2	2ms
Task3	4ms
Task4	8ms
Task5	16ms
Task6	32ms
Task7	64ms

Tabla 2. Periodicidad de las tareas

Offsets

Para el cálculo del retardo se utiliza un algoritmo de enmascaramiento que evita convergencia en la ejecución de tareas. Por ejemplo, para manejar 7 tareas, se necesitan 6 “*offsets*” o retardos. Para el cálculo de los mismos se utilizan las siguientes reglas:

$$Offset_{TaskN} = 2^{(N-1)} \quad (3.1)$$

$$Mask_{TaskN} = 2^N - 1 \quad (3.2)$$

Enmascaramiento mínimo para tareas y longitud de conteo

Considerando que se tienen N tareas que se pueden traslapar, es necesario seleccionar un contador del tipo que soporte todas las tareas, con la relación de que para N tareas se necesitan al menos N bits, en el caso de 7 tareas con un contador de 8 bits es suficiente. Sin embargo, para la implementación del código decidimos emplear 32 bits para tener la facilidad de declarar más tareas. La siguiente tabla muestra la relación entre la máscara asignada y el número de tarea:

TaskNumber	1	2	3	4	5	6	7
TaskMask	1	3	7	15	31	63	127

Tabla 3. Máscaras de las tareas para el scheduler.

3.2.2. Tiempo base para el SO

Para el sistema operativo seleccionado es necesario contar con una base de tiempo, conocida como “*OS Tick Time*”, la cual debe ser una fracción de la tarea periódica menor, y cuya característica principal es que debe ser un mínimo común múltiplo de todos los periodos de las tareas a incluir. Esta característica permite que la siguiente ecuación sea válida para la ejecución a tiempo de todas las tareas:

$$TaskRate = OsTickTime * (TaskMask + 1) \quad (3.3)$$

Si conocemos la máscara actual y el periodo de ejecución, entonces se puede calcular el tiempo base requerido para cumplir con la postulación de la tarea:

$$OsTickTime = \frac{TaskRate}{TaskMask + 1} \quad (3.4)$$

Como sabemos que el tiempo mínimo para la tarea con el periodo más corto es de 1 ms, el tiempo base seleccionado es:

$$Task1_{period} = 1ms \gg OsTickTime = 500\mu s \quad (3.5)$$

3.2.3. Tabla de verdad para el algoritmo

Con el fin de probar las propuestas anteriores, se muestra la Tabla 4, donde se puede observar que algoritmo propuesto funciona adecuadamente según la máscara especificada considerando los retardos y las cuentas. Por ejemplo, la tarea uno que se ejecuta cada milisegundo se activa cada que vez que el bit menos significativo del contador es igual a cero. Este resultado es lógico considerando que la resta entre la máscara de la tarea 1 y el offset es igual a cero, más aún, como la máscara únicamente tiene valor en el bit0 (el menos significativo), el resultado de la operación “and” siempre será cero cuando el contador sea cero. Así por ejemplo para la tarea 4 que se ejecuta cada 8 milisegundos, la resta entre la máscara y el offset es igual a siete, por lo que la tarea siempre será activada cuando los tres bits menos significativos del contador sean iguales a 1.

TaskNumber	-	1	2	3	4	5	6	7	8	9	TaskPeriod:	OSTick * (TaskMask + 1)
TaskPeriod	-	0.001	0.002	0.004	0.008	0.016	0.032	0.064	0.128	0.256	Condition:	(Counter & MASK) == (MASK - Offset)
TaskMask	0	1	3	7	15	31	63	127	255	511	OSTick Time:	0.0005
Counter	-	000000001	000000011	000000111	000001111	000011111	000111111	001111111	011111111	111111111	MinTaskTime:	0.001
0	000000000	OFS = 1										
1	000000001		OFS = 2									
2	000000010	OFS = 1										
3	000000011			OFS = 4								
4	000000100	OFS = 1										
5	000000101		OFS = 2									
6	000000110	OFS = 1										
7	000000111				OFS = 8							
8	000001000	OFS = 1										
9	000001001		OFS = 2									
10	000001010	OFS = 1										
11	000001011			OFS = 4								
12	000001100	OFS = 1										
13	000001101		OFS = 2									
14	000001110	OFS = 1										
15	000001111					OFS = 16						

Then in order to support N tasks with out overlapping

$$Mask_{TaskN} = 2^N - 1$$

$$Offset_{TaskN} = 2^{(N-1)}$$

Tabla 4. Algoritmo de distribución de las tareas.

3.2.4. Estructura de archivos

En esta sección se muestran las consideraciones tomadas durante el diseño, así como los diagramas empleados para el Manejador de Tareas (TM). El TM utiliza estructuras de control definidas y temporizadores de propósito general. El TM es ejecutado en cada interrupción del (“Tick Timer”), con el fin de establecer la tarea que será despachada. La estructura de archivos realizada queda como se muestra en el siguiente diagrama.

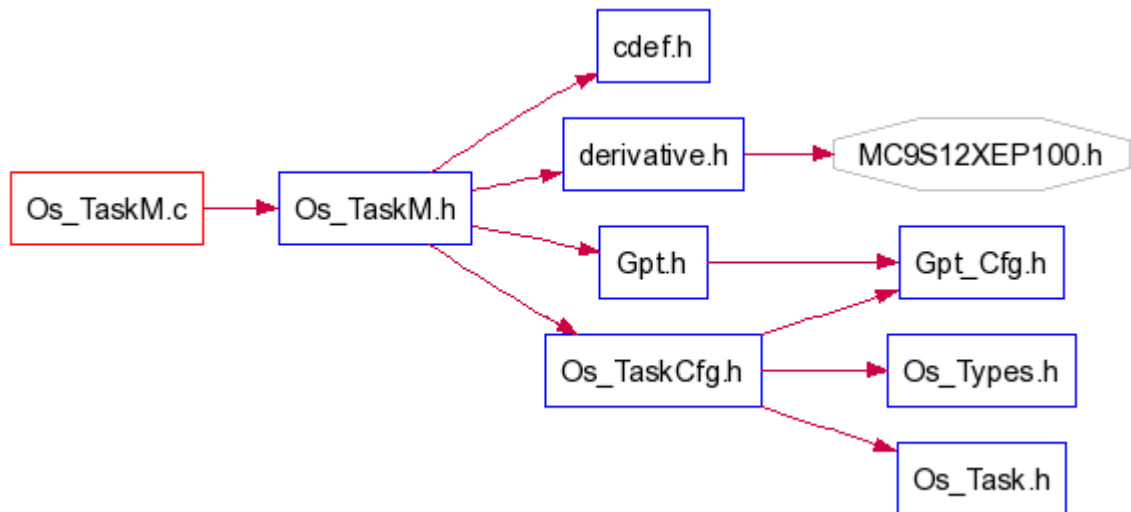


Figura 8. Estructura de archivos del sistema operativo

3.2.5. Datos miembro de la estructura del TM

Los datos miembro utilizados en la estructura de configuración son como se muestra en el siguiente diagrama.

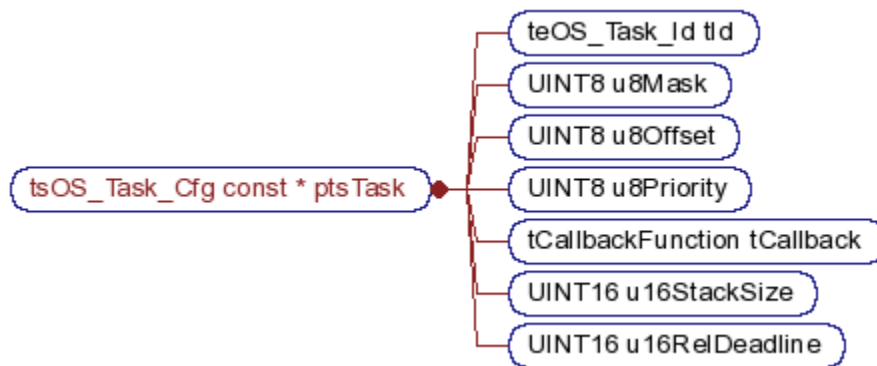


Figura 9. Miembros de la estructura de configuración de las tareas.

Estos datos describen por completo la información utilizada para la implementación del manejo funcional del despachador del Sistema Operativo. Todas las tareas son inicializadas en arreglos con valores constantes, incluyendo la prioridad, que debe ser única. Esto podría ser una limitación para el despachador de tareas por el fenómeno de inversión de prioridades. Sin embargo, una estructura dinámica que registra el estado de las tareas se ha incluido para eliminar esta limitante. Esto se logra realizando unos cambios simples generando la estructura que se muestra en el siguiente diagrama:

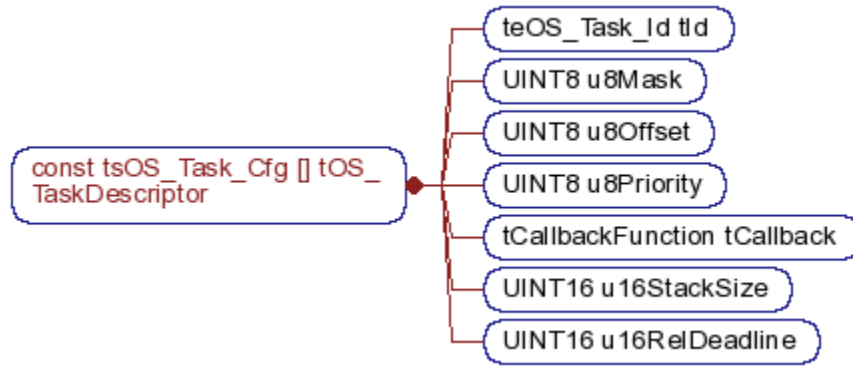


Figura 10. Arreglo de la estructura de configuración de las tareas.

La estructura propuesta para el bloque de control de tareas, TCB (*Task Control Block*) por sus siglas en inglés, incluye un apuntador a la estructura que describe los parámetros de la tarea, el estado de está y el número de activaciones pendientes, como se muestra en el siguiente diagrama.

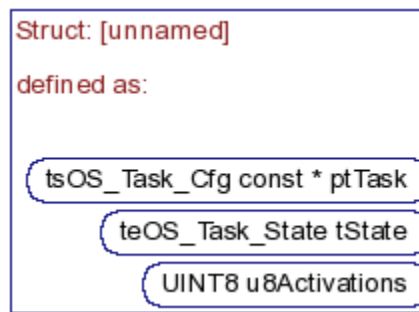


Figura 11. Miembros de la estructura del bloque de control de tareas

3.2.6. Diseño de la estructura del Buffer

El buffer utilizado para almacenar las tareas que están en espera de ser despachadas se constituye de una estructura circular. El buffer se conforma de una estructura de estado que incluye dentro de sus parámetros miembro, un apuntador al buffer del índice de tareas, un apuntador que indica la ubicación donde se puede enfilear una tarea cuyo estado ha sido cambiado a “*Ready*” y un apuntador que “apunta” a la siguiente tarea a ser ejecutada. La rutina para obtener el identificador de la tarea del TCB (*Task Control Block*) en turno es una función implementada a través de una macro para su acceso. Finalmente se utiliza un contador para llevar el registro del número de activaciones de la tarea enésima. El diagrama funcional se muestra a continuación.

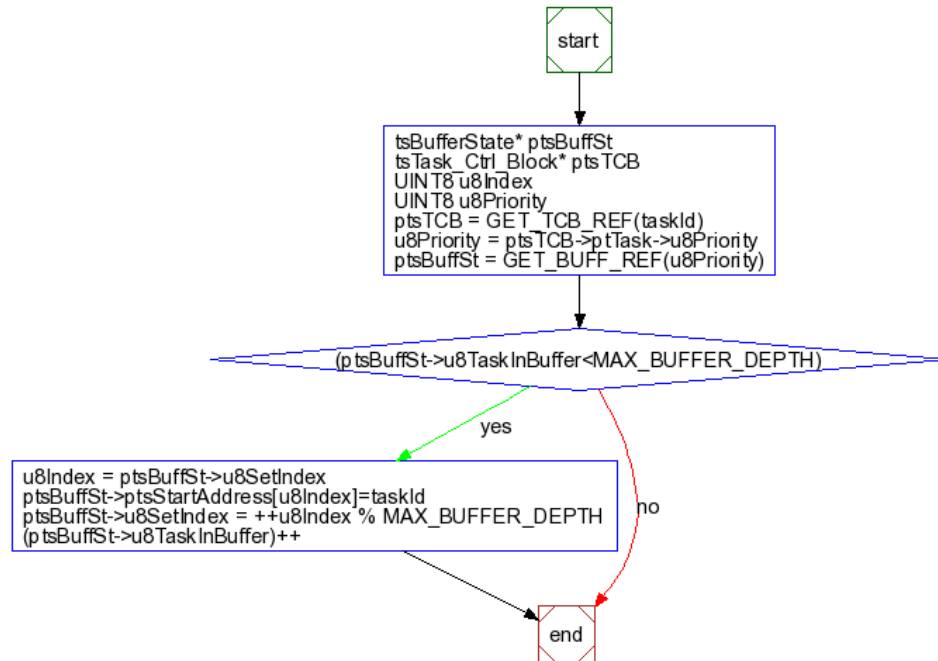


Figura 12. Rutina de activación de tareas

3.2.1. Funcionamiento del TCB

La función “*Get_Task_Id*” busca la tarea enfilada más antigua esperando a ser ejecutada, comenzando la búsqueda desde el buffer de mayor prioridad. La implementación del contador en la estructura de estado del buffer permite acelerar la búsqueda. Esto es, si una tarea se encuentra en espera, la función retorna su valor de identificación, de lo contrario se retorna el mensaje de “*No_Task*”. La descripción de este proceso se muestra en el diagrama de la Figura 13.

3.2.2. Guardar y Restaurar el contexto de la Tarea

Con el fin de guardar y restaurar el contexto de la tarea en ejecución, se hace uso del mecanismo de interrupción por hardware. Cuando una interrupción ocurre, los componentes de hardware del microcontrolador almacenan los registros del CPU en la pila. Cuando se ejecuta la instrucción RTI, nuevamente el hardware se encarga de cargar en el CPU los datos almacenados previamente en la pila.

A través de este método, cuando ocurre una interrupción los datos almacenados en la pila son copiados en una ubicación específica de memoria reservada para este fin, la cual ha sido asociada previamente con la tarea en ejecución. Después de éste proceso la interrupción es ejecutada, una vez terminada la ejecución, la función del despachador es llamada para restablecer el contexto de la tarea. Para llevar a cabo esta funcionalidad la información de la pila almacenada es insertada nuevamente en la pila principal, al ejecutar la instrucción RTI, el hardware carga los datos en el registro del CPU como lo haría normalmente al retornar de una interrupción cualquiera. Este proceso permite aprovechar las implementaciones en

hardware del microcontrolador para la rápida respuesta a un cambio de tarea durante la ejecución de un sistema operativo tipo “preemptive” y está descrito en el diagrama de la Figura 14.

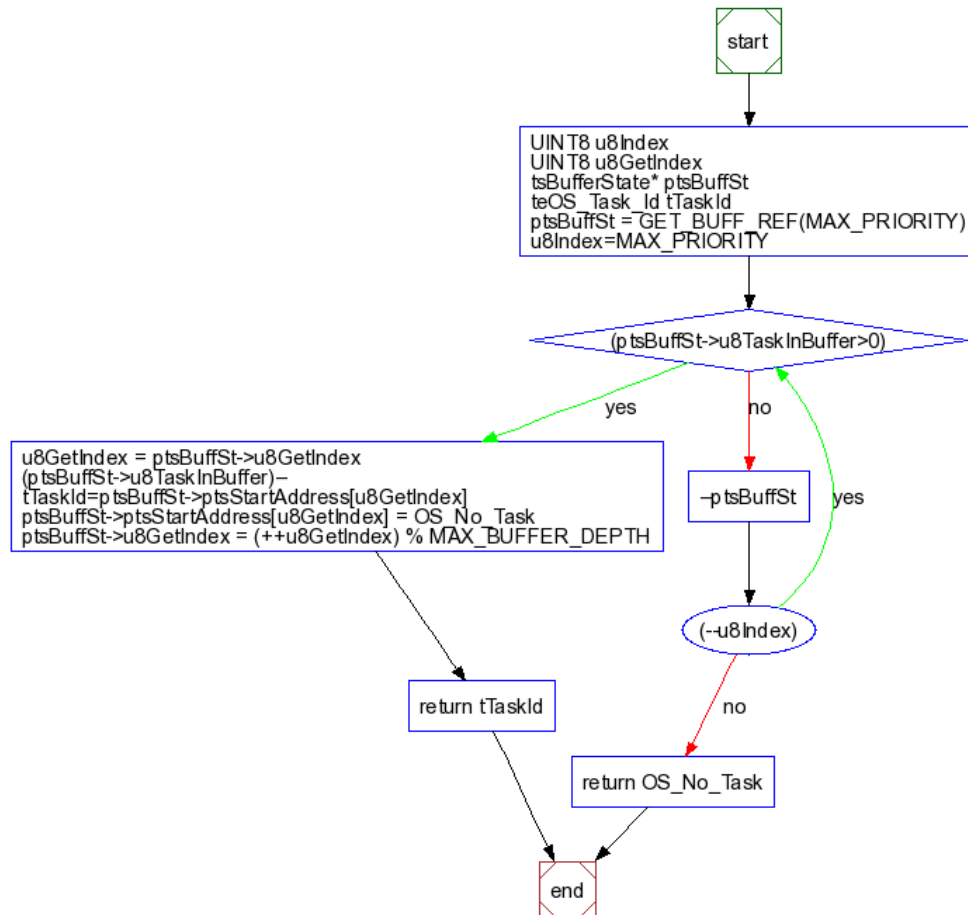


Figura 13. Diagrama a bloques de la rutina para ejecutar tareas.

También es posible que una tarea sea activada por otra de mayor prioridad. Cuando esto ocurre la tarea actual debe ser interrumpida. En este caso se utiliza una interrupción por software de tal forma que el hardware almacena los registros del CPU en la pila principal, estableciendo las mismas condiciones para el manejo del contexto que en el caso de una interrupción externa.

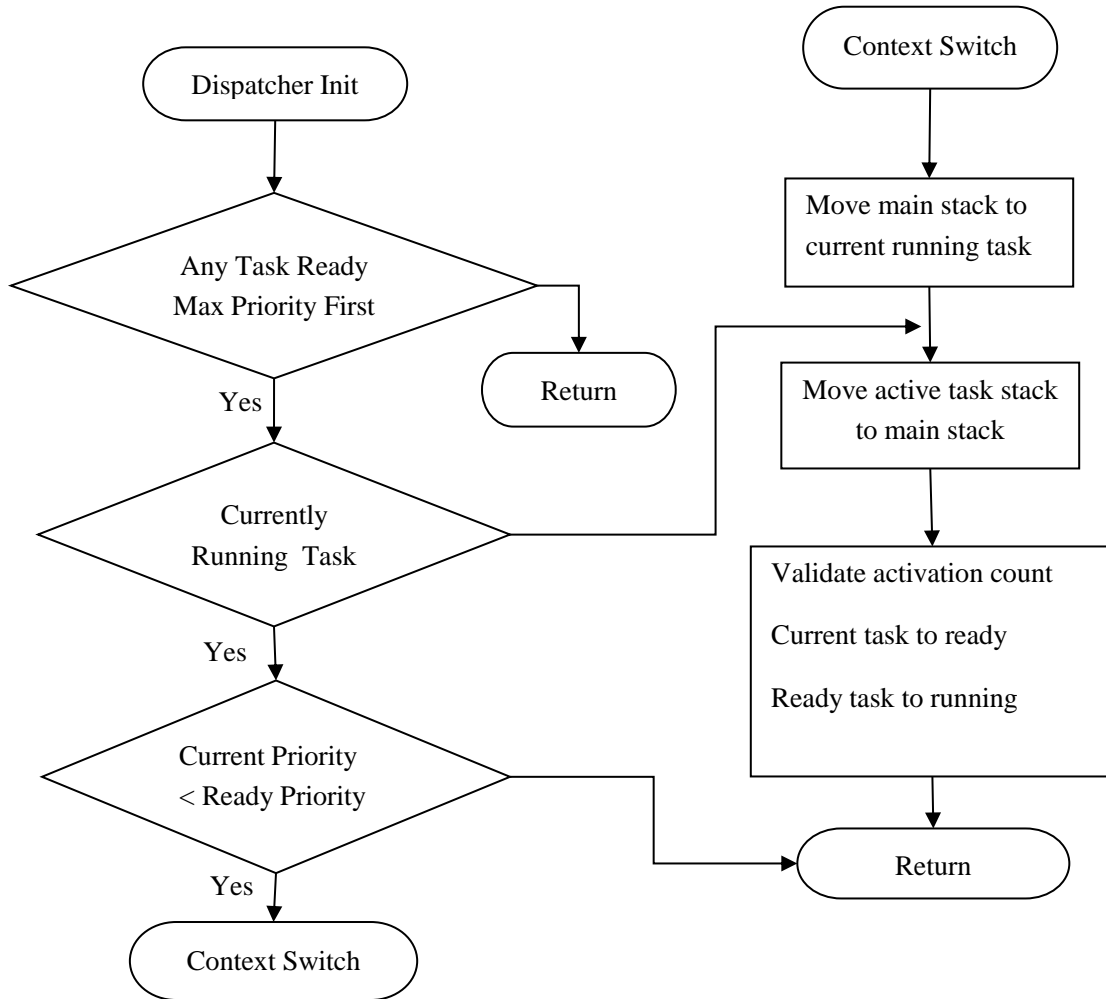


Figura 14. Secuencia de ejecución durante un cambio de contexto.

3.3. Manejador de Mensajes de CAN

El proceso de comunicación es llevado a cabo utilizando el protocolo CAN, el cual en su capa física es implementado utilizando medios alámbricos de conductor eléctrico. En su capa de enlace se aprovecha el módulo embebido de CAN que provee la tarjeta DEMO9S12XEP100, así como la configuración de filtros para la capa de transporte. Continuando con la implementación por capas es necesario realizar un manejador que tome los datos de la capa de transporte y llame las funciones que requieren de esos datos en la capa de aplicación. Derivado de esta estructura de operación es que el manejador de mensajes implementado requiere de una rutina de lectura periódica, como se muestra en el diagrama de la Figura 15. Cuando se ha recibido un nuevo mensaje el proceso a seguir consta de limpiar la bandera de recepción, el contador de retardos, seguido de la bandera de retardo y recuperación de los datos del mensaje.

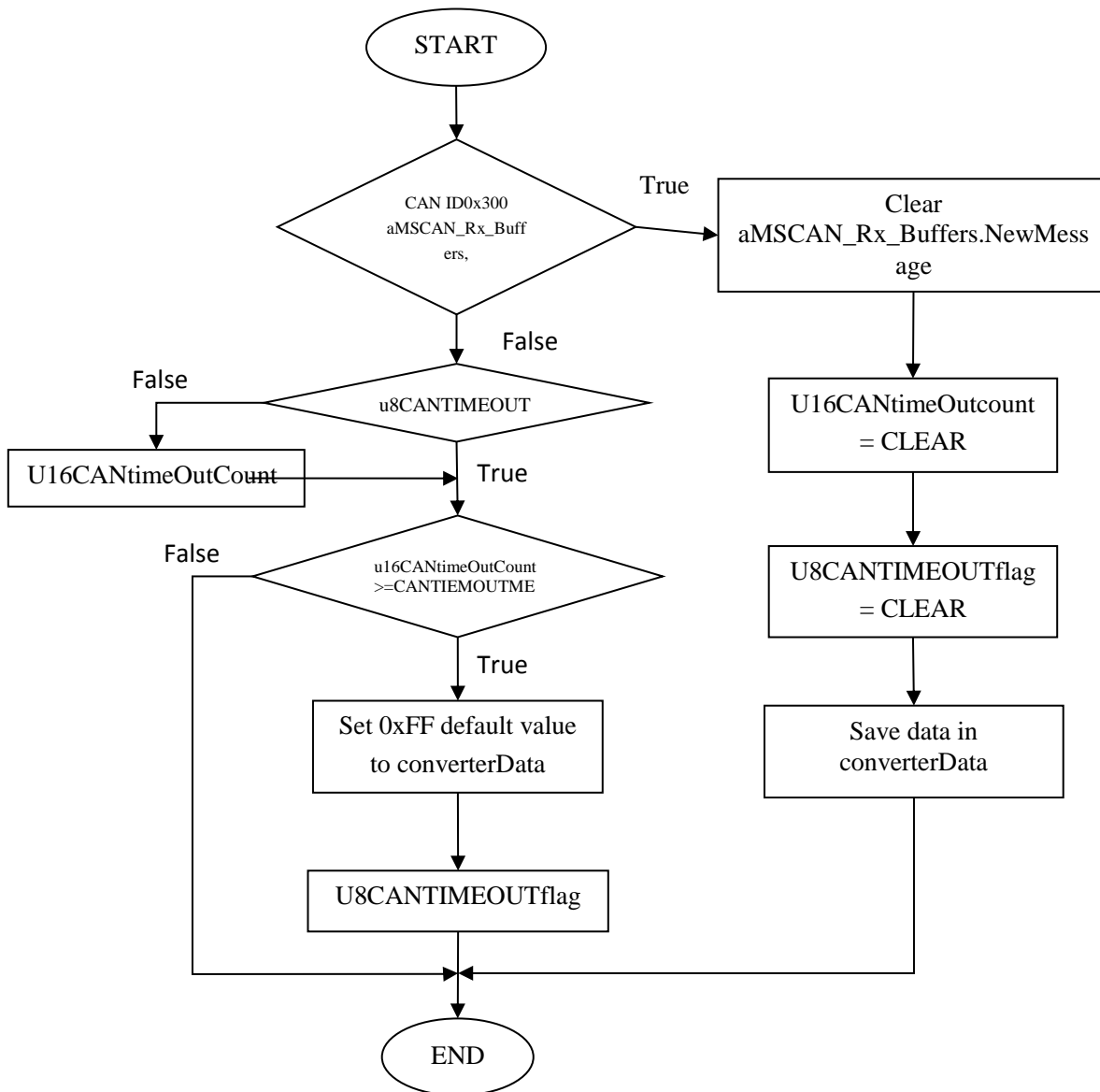


Figura 15. Rutina de operación del manejador de mensajes de CAN

Con este proceso de obtención de mensajes la aplicación puede determinar si el mensaje recibido es útil para mantener el convertidor en operación. En caso de que no se haya recibido un mensaje nuevo con la instrucción de “*stay alive*”, se incrementa el contador de retardos hasta alcanzar el tiempo límite para el retardo total, en este momento se configuran los valores por defecto para los parámetros del convertidor y se activa la bandera de retardo. Este proceso es eliminado cuando se recibe un mensaje con datos válidos.

3.4. Transmisión de Mensajes CAN

Para la transmisión de mensajes de CAN se utiliza un periodo base, el cual al ser alcanzado reporta el estado de operación del voltaje de salida del convertidor. Este comportamiento se describe en el diagrama de la Figura 16.

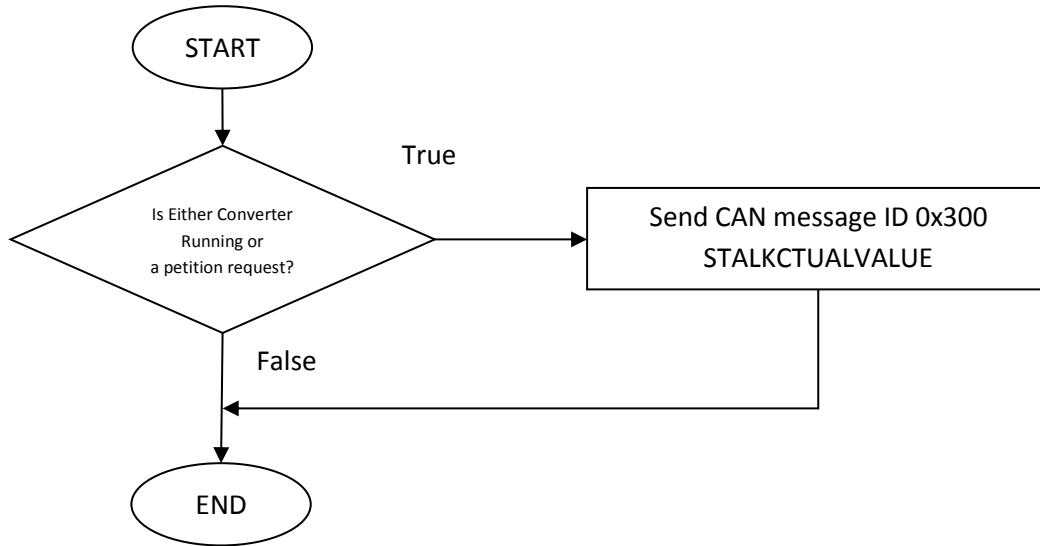


Figura 16. Diagrama de transmisión de mensajes por CAN

3.5. Circuito Convertidor

El circuito para el convertidor fue propuesto por Liang. et. al. (3), este se muestra en la Figura 17.

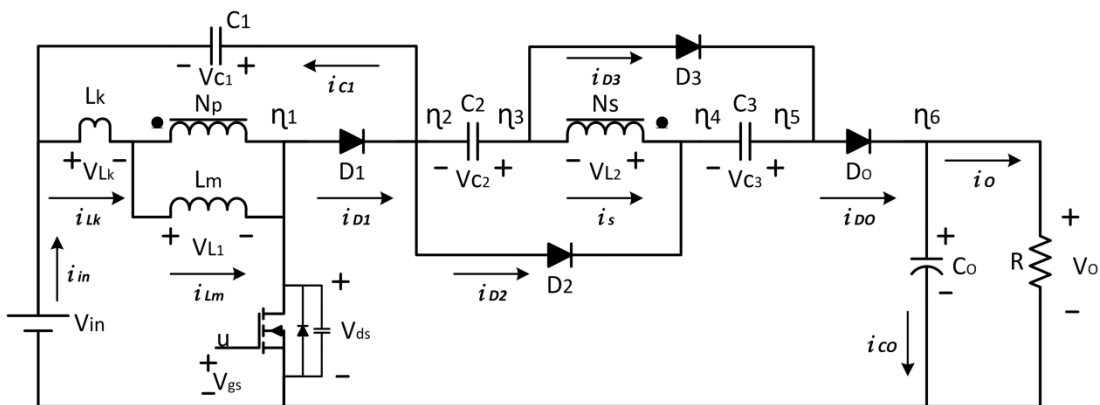


Figura 17. Topología del convertidor de elevación de alta ganancia.

La fuente de alimentación del circuito se compone de una batería ácido-plomo de 12 V, el voltaje de entrada se identifica por V_{in} , el conmutador es un Mosfet de SiC cuya variable de activación es u , el

inductor acoplado N_p y N_s , diodos D_x y capacitores C_x . El modelo del inductor incluye la inductancia de fuga y la inductancia de magnetización.

Basándose en la topología, el convertidor combina las técnicas de capacitor conmutado e inductor conmutado. La técnica del capacitor conmutado propone que los capacitores pueden ser cargados en paralelo y descargados en serie para lograr el alto nivel de ganancia. Tomando este concepto, los capacitores C_2 y C_3 se colocan en el secundario del inductor. De esta forma los capacitores C_2 y C_3 son cargados en paralelo y descargados en serie por el inductor del secundario cuando el conmutador es apagado y encendido. Debido a que el voltaje en los capacitores puede ser ajustado en base a la relación del inductor acoplado, se puede conseguir una alta ganancia de elevación.

El principio de operación se basa principalmente en el período en que se encuentra encendido el conmutador, es durante este lapso de tiempo que el inductor acoplado induce voltaje en el secundario y la inductancia de magnetización capacitores L_m es cargada por capacitores V_{in} . El voltaje induce provoca que V_{in} , V_{C1} , V_{C2} y V_{C3} entreguen su energía a la salida en serie. El inductor acoplado se utiliza como un transformador en el convertidor. Cuando el conmutador se encuentra en estado de apagado, la energía de magnetización del inductor L_m se entrega al secundario a través del acoplamiento del inductor para cargar los capacitores C_2 y C_3 en paralelo, el inductor es utilizado como un transformador en un convertidor con topología “flyback”.

El modelo que describe el comportamiento del sistema es de estructura variable, sus ecuaciones que la describen son como sigue:

Sean:

$x_1: V_{\eta_1}$	Voltaje del Mosfet
$x_2: V_{\eta_2}$	Voltaje en el cátodo de D1
$x_3: V_{\eta_3}$	Voltaje en el ánodo de D3
$x_4: V_{\eta_6}$	Voltaje en el Capacitor C0
$x_5: V_{C3} = V_{\eta_5} - V_{\eta_4} = x_9 - x_8$	Voltaje en el Capacitor C3
$x_6: i_{Lm}$	Corriente en la Inductancia magnetizante
$x_7: i_{Lk}$	Corriente en la Inductancia de fuga
$x_8: V_{\eta_4}$	Voltaje en el cátodo de D2
$x_9: V_{\eta_5}$	Voltaje en el ánodo de D0

Aplicando la ley de corrientes de Kirchhoff para el nodo η_x se tiene:

$$\eta_1: \quad V_{in} - x_1 = k(V_{in} - x_1) + L_k \dot{x}_7 \quad (3.6)$$

$$x_7 = D(x_1, x_2) + \frac{x_1}{R_{MOS}} \delta_{MOS} + C_{MOS} \dot{x}_1$$

$$\eta_2: \quad D(x_1, x_2) + C_2(\dot{x}_3 - \dot{x}_2) = D(x_2, x_8) + C_1(\dot{x}_2 - \dot{V}_{in}) \quad (3.7)$$

$$\eta_3: \quad D(x_3, x_9) + C_2(\dot{x}_3 - \dot{x}_2) + \frac{x_6}{n} = 0 \quad (3.8)$$

$$\eta_4: \quad D(x_2, x_8) + C_3(\dot{x}_9 - \dot{x}_8) + \frac{x_6}{n} = 0 \quad (3.9)$$

$$\eta_5: \quad D(x_3, x_9) = C_3(\dot{x}_9 - \dot{x}_8) + D(x_9, x_4) \quad (3.10)$$

$$\eta_6: \quad D(x_9, x_4) = \frac{x_4}{R} + C_0 \dot{x}_4 \quad (3.11)$$

Realizando la simplificación algebraica, las ecuaciones de estado que describen el comportamiento del sistema son:

$$\dot{x}_1 = \frac{1}{C_{MOS}} \left(x_6 + k \cdot n(D_0 - D_2 - D_3) - D_1 - \frac{x_1}{R_{MOS}} \delta_{MOS} \right) \quad (3.12)$$

$$\dot{x}_2 = \frac{1}{C_1} (D_1 - D_0) \quad (3.13)$$

$$\dot{x}_3 = \frac{1}{C_1} (D_1 - D_0) - \frac{1}{C_2} (D_0 - D_2) \quad (3.14)$$

$$\dot{x}_4 = \frac{1}{C_0} \left(D_0 - \frac{x_4}{R} \right) \quad (3.15)$$

$$\dot{x}_5 = D_0 - D_3 \quad (3.16)$$

$$\dot{x}_6 = \frac{k(V_{in} - x_1)}{L_m} \quad (3.17)$$

donde $y = x_4$, D_i es la corriente que fluye por el diodo i-esimo, donde: $D_0(a_2, x_4)$, $D_1(x_1, x_2)$, $D_2(x_2, a_1)$, $D_3(x_3, a_2)$, son funciones que dependen de los estados. La función D es:

$$D(w, z) = \frac{w - z - V_T}{R_T} \delta, \quad \delta = \begin{cases} 0, & (w - z - V_T) < 0 \\ 1, & (w - z - V_T) \geq 0 \end{cases} \quad (3.7)$$

Para

$$\begin{aligned} a_1 &= n \cdot k(V_{in} - x_1) + x_3 \\ a_2 &= a_1 - \frac{x_5}{C_3} \end{aligned} \quad (3.8)$$

3.6. Controlador del Convertidor

Para la realización del controlador del convertidor se realizaron algunas modificaciones al circuito original, quedando el nuevo circuito como se muestra en la Figura 18. Estas modificaciones se realizan para tener una medición directa de la corriente del inductor I_L , la corriente en el capacitor de salida i_{CO} y el voltaje de salida V_O . Para la realización del controlador se utiliza una técnica conocida como control por modos deslizantes en una superficie de deslizamiento controlada por corriente (6). Para la realización de este controlador se plantea la referencia para la corriente como:

$$i_{ref} = K[V_{ref} - v_o] \quad (3.12)$$

donde V_{ref} y v_o denotan la referencia y el voltaje instantáneo de salida respectivamente; β denota la relación entre el voltaje de salida y la interfaz del sensor; K es la ganancia de amplificación del voltaje de error. El valor de K suele ser de gran magnitud para mejorar la respuesta dinámica y minimizar el error en estado estacionario. Se pretende que $I_{ref(SS)} = I_{L(SS)} = K[V_{ref} - v_{o(SS)}]$. La superficie deslizante es:

$$S = \alpha_1 \xi_1 + \alpha_2 \xi_2 + \alpha_3 \xi_3 \quad (3.13)$$

donde los coeficientes α_i representan los coeficientes de deslizamiento. La función de conmutación es $u = \frac{1}{2}(1 - \text{sign}(S))$. Las variables de control son el estado ξ_1 como la corriente de error, ξ_2 el voltaje de error, y ξ_3 como la integral de la corriente y voltaje de error, los cuales se expresan como:

$$\begin{cases} \xi_1 = i_{ref} - i_L \\ \xi_2 = V_{ref} - v_o \\ \xi_3 = \int [\xi_1 + \xi_2] dt \end{cases} \quad (3.14)$$

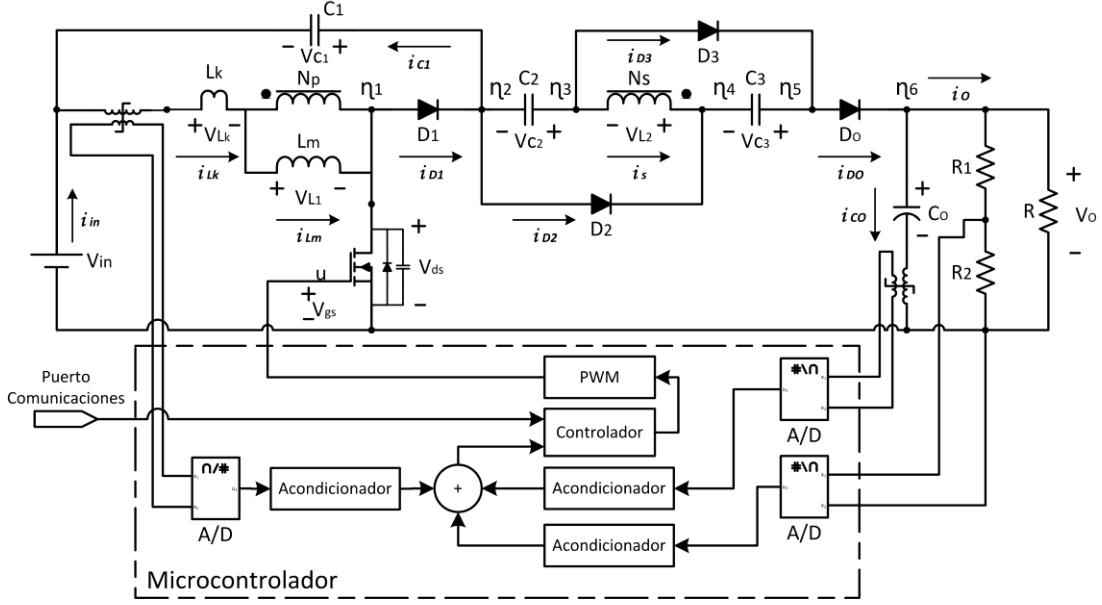


Figura 18. Interfaz electrónica del convertidor con el sistema de control.

Derivando las ecuaciones de (3.14) se obtiene:

$$\begin{cases} \dot{\xi}_1 = \frac{v_o - (1+n) \cdot V_{in}}{L_m(2n+1)}(1-u) + \frac{nV_{in}}{L_m} \cdot u \\ \dot{\xi}_2 = -\frac{1}{C_0} i_{C_0} \\ \dot{\xi}_3 = (i_{ref} - i_L) + (V_{ref} - v_o) \\ = (K+1)[V_{ref} - v_o] - i_L \end{cases} \quad (3.15)$$

las condiciones de existencia son:

$$\begin{cases} \frac{v_{o(SS)}}{(n+1)} - K_1 i_{C_0(MIN)} + K_2 [V_{ref} - v_{o(SS)}] - K_3 i_{L(MIN)} < V_{in(MIN)} \\ V_{in(MIN)} - K_4 i_{C_0(MAX)} + K_5 [V_{ref} - v_{o(SS)}] - K_3 i_{L(MAX)} > 0 \end{cases} \quad (3.16)$$

con:

$$K_1 = \frac{L_m(2n+1) \alpha_2}{C_0(1+n) \alpha_1}, \quad K_2 = \frac{L_m(2n+1)(K+1) \alpha_3}{(1+n) \alpha_1}, \quad K_3 = \frac{L_m(2n+1) \alpha_3}{(1+n) \alpha_1}$$

$$K_4 = \frac{L_m \alpha_2}{C_0 n \alpha_1}, \quad K_6 = \frac{L_m(K+1) \alpha_3}{n \alpha_1}, \quad K_6 = \frac{L_m \alpha_3}{n \alpha_1}$$

El control equivalente queda como:

$u = \gamma \left[\frac{L_m(2n+1) \alpha_2}{C_0} i_{C_0} - \frac{\alpha_3 L_m(2n+1)}{\alpha_1} \left((K+1)[V_{ref} - v_o] - i_L \right) - v_o + (1+n) \cdot V_{in} \right]$	(3.17)
$\gamma = \frac{1}{n(2n+1) - v_o + (1+n) \cdot V_{in}}$	

3.7. Implementaciones

En esta sección se muestran algunos extractos de la implementación según la sección dedicada al diseño. Inicialmente se muestra la implementación del algoritmo de planeación. Para la configuración de las tareas se utiliza la variable *ser_tSCHM_configuration* la cual tiene la estructura mostrada en la figura 19. Para la implementación de la configuración del tiempo base, se debe recurrir a la sección de configuración del GPT (*General Purpose Timer*) en el canal 0, pues este es el seleccionado para ejecutar la interrupción del tiempo base asociada con la rutina de planeación de tareas.

En la configuración del planificador se muestra una lista de las tareas soportadas, donde cada tarea tiene propiedades configurables, como el estado de la tarea. Todas las tareas inician en el estado de “*suspended*”, lo que significa que estas tareas serán propuestas para su ejecución durante la planificación. Debe tenerse precaución pues si la tarea es inicializada con la etiqueta “*Task disabled*” esta tarea nunca será ejecutada.

3.7.1. Algoritmo de planeación

La operación del planificador o “*scheduler*” puede ser dividido en tres partes principales:

- **Inicialización:**
En esta sección la configuración del planificador es ejecutada, el temporizador del SO es inicializado, el contador general y las variables locales son inicializadas a sus valores por defecto.
- **Tarea Background:**
Esta es la tarea que se ejecuta continuamente cuando ninguna tarea requiere de ejecución. Se considera como una tarea más con la prioridad más baja. Esta tarea es parte del ciclo infinito de ejecución de tareas, que en esencia ejecuta una tarea que no realiza actividades del sistema, excepto si es requerido, estimaciones de la carga de trabajo del micro o las tareas.
- **Manejador de la rutina de tiempo base:**
Esta es una función encargada de incrementar el contador general y activar las tareas cuyo periodo de ejecución corresponde al contador de tiempo actual. Este manejador es llamado cada vez que se ejecuta una interrupción del tiempo base.

```

31  tsSCHM_Cfg ser_tSCHM_configuration =
32  {
33      /* GPI Channel 0 */
34      {
35          {
36              PIT0,                /* Channel Id */
37              125,                /* Period (micro seconds) */
38              PIT_Type_HihgResolution, /* Channel type: HihgRes or HihPeriod */
39              SchM_OsTick         /* Interruption Callback */
40          },
41          TRUE,                  /* Shall be initialized */
42          TRUE,                  /* Shall be started */
43          TRUE                   /* Shall its interruption be enabled */
44      },
45      /* Tasks Configuration*/
46      {
47          /* Task 1 */
48          {
49              SCHM_Task_1,      /* Task Id */
50              7,               /* Task Mask */
51              0,               /* Task Offset */
52              SchM_Task_1,     /* Task Callback */
53              SCHM_Task_Suspended /* Task Status */
54          },
55          /* Task 2 */
56          {
57          },
58          /* Task 3 */
59          {
60          },
61          /* Task 4 */
62          {
63          },
64          /* Task 5 */
65          {
66          },
67          /* Task 6 */
68          {
69          },
70          /* Task 7 */
71          {
72          },
73          /* Task 8 */
74          {
75          },
76          /* Task 9 */
77          {
78          },
79          /* Task 10 */
80          {
81          },
82          /* Task 11 */
83          {
84          },
85          /* Task 12 */
86          {
87          },
88          /* Task 13 */
89          {
90          },
91          /* Task 14 */
92          {
93          },
94          /* Task 15 */
95          {
96          },
97          /* Task 16 */
98          {
99          },
100         /* Task 17 */
101         {
102         },
103     }
104 };

```

Figura 19. Implementación del algoritmo de planeación.

En la figura 20 se muestra el diagrama funcional del algoritmo de planeación de tareas. Se tuvo las siguientes consideraciones sobre la implementación del planificador de tareas:

- Soporte de hasta 31 tareas.
- La periodicidad de las tareas es controlada a través de la configuración del periodo de tiempo base en la estructura de configuración general.

- Se trata de un planificador tipo “preemptive”, por lo que las tareas son interrumpidas y ejecutadas según su prioridad.

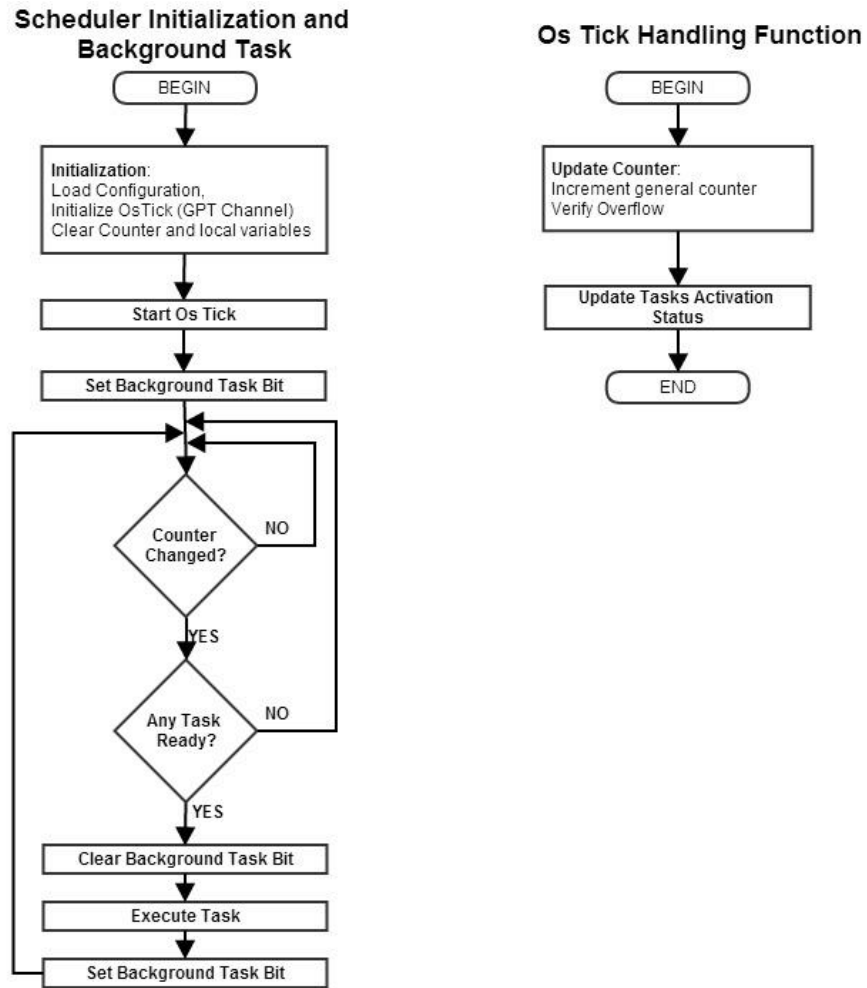


Figura 20. Diagrama a bloques del algoritmo de planeación de tareas.

- El tiempo total de duración de ejecución de una tarea debe ser menor que el tiempo base, de otra forma se puede generar un retardo adicional en la ejecución de las tareas.
- Cada vez que una tarea es ejecutada se puede asignar un pin de ejecución Esto es útil para revisar la carga, o tiempo requerido para la ejecución de cada tarea. Por ejemplo, se puede realizar la siguiente asignación:

<i>Task</i>	<i>Output</i>
Background	PB0
Task1	PB1
Task6	PB2
Task9	PB3
Task21	PB4

Tabla 5. Tabla de terminales para medir la carga de una tarea.

- La implementación actual es capaz de establecer el origen de dos posibles fallas durante el tiempo de ejecución, estas son:
 1. La ejecución de una tarea retrasada
 2. Transición errónea. Esto ocurre cuando una tarea debe ser activada pero su estado previo no es “suspended”.

3.7.2. Implementación del activador de tareas.

La función de activación es ejecutada con cada interrupción de la base de tiempo, utiliza la información del TCB para determinar si se ha alcanzado el máximo número de activaciones, si esto es verdadero el estado actual de la tarea es cambiado a “Ready”, y es guardado en el buffer correspondiente. Si el número máximo de activaciones ha llegado a su límite se regresa un mensaje de error. El diagrama correspondiente se muestra a continuación.

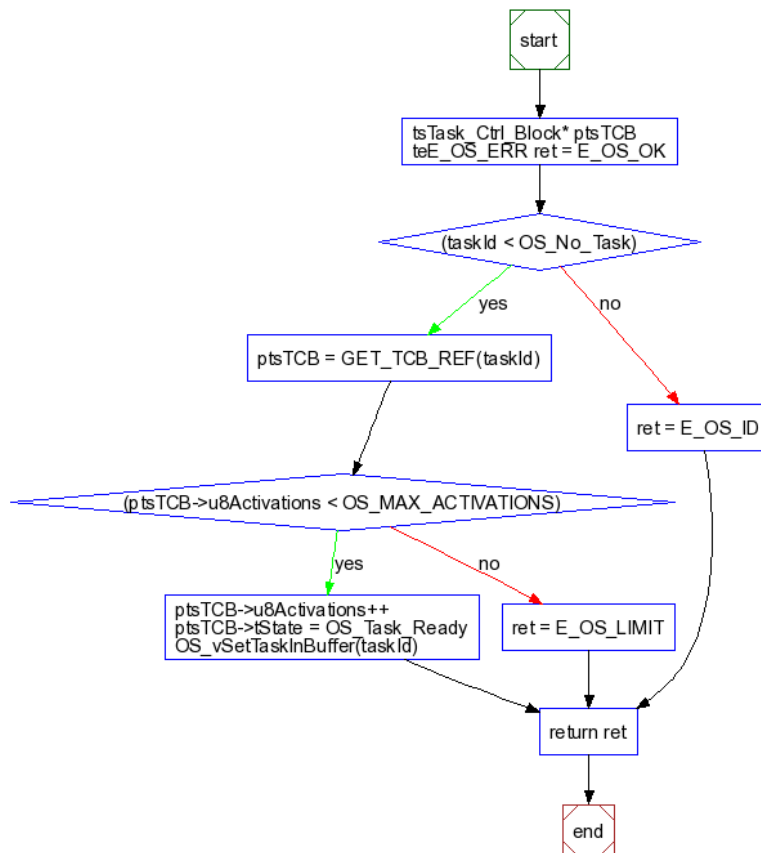


Figura 21. Diagrama de secuencia del activador de tareas.

3.7.3. Descripción del despachador de funciones.

El despachador obtiene el identificador de tarea más antiguo que se haya almacenado en el buffer con el fin de ejecutar la tarea. Si el proceso de activación no sufrió falla alguna, el despachador se encarga de cambiar su estado a “Run”, a través del parámetro “status” del bloque de control de tareas (TCB). Si

ninguna función esta activada para ser ejecutada el despachador regresa la etiqueta “No_Task”. El diagrama del comportamiento del despachador se muestra a continuación.

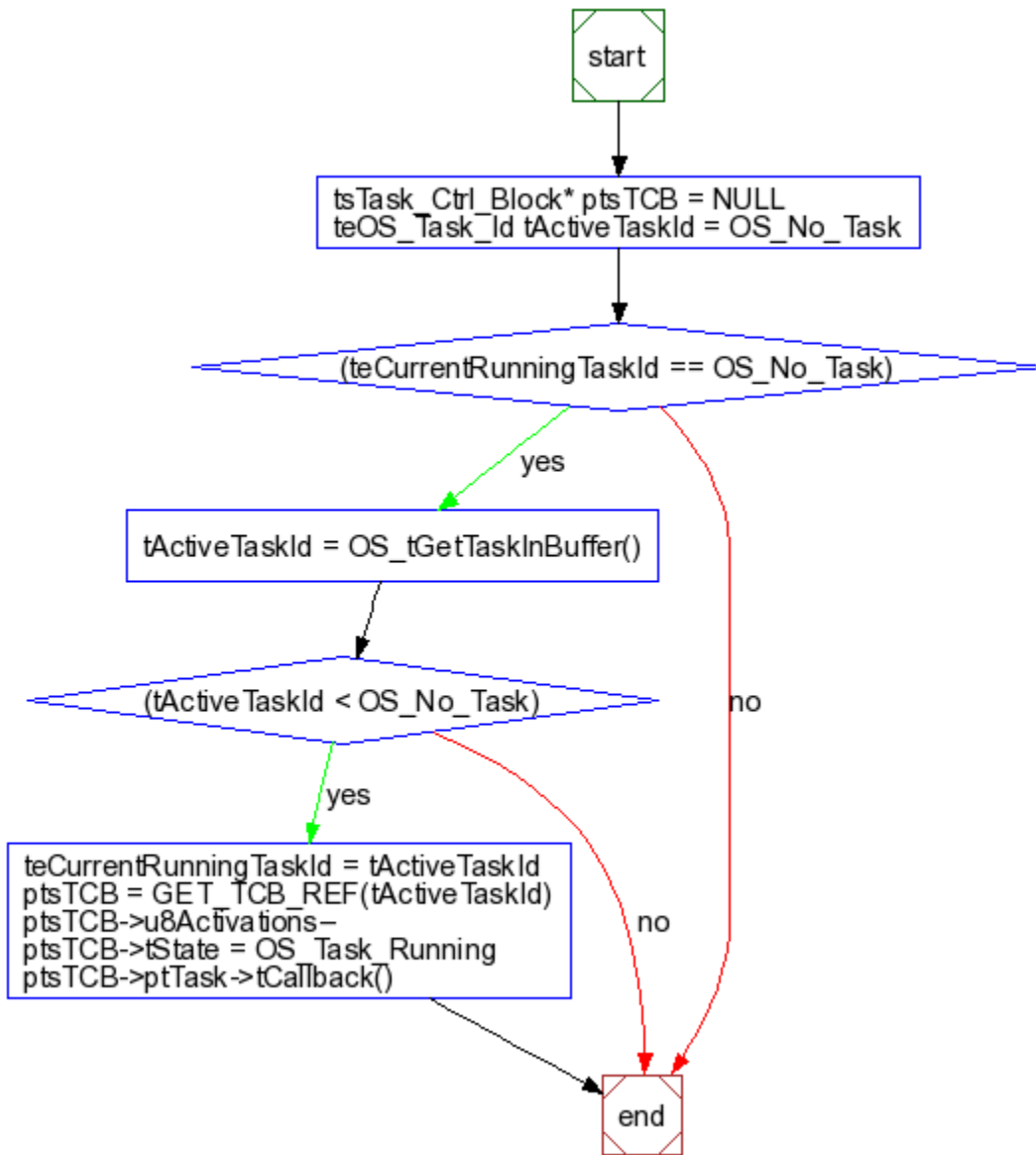


Figura 22. Diagrama de secuencia del despachador de tareas.

3.7.4. Descripción de la función de terminación

Finalmente, la función de terminación incluida en cada tarea como medio para permitir al Sistema Operativo dar por concluida la ejecución de una tarea y continuar con su rutina. Cada función debe ser incluida al final de su rutina. El trabajo principal de esta función es cambiar el estado de la tarea a “suspend”. Para conocer cual tarea ha terminado se utiliza una variable global que guarda el valor del

identificador de la tarea actual. El diagrama de comportamiento de esta función es como se muestra a continuación.

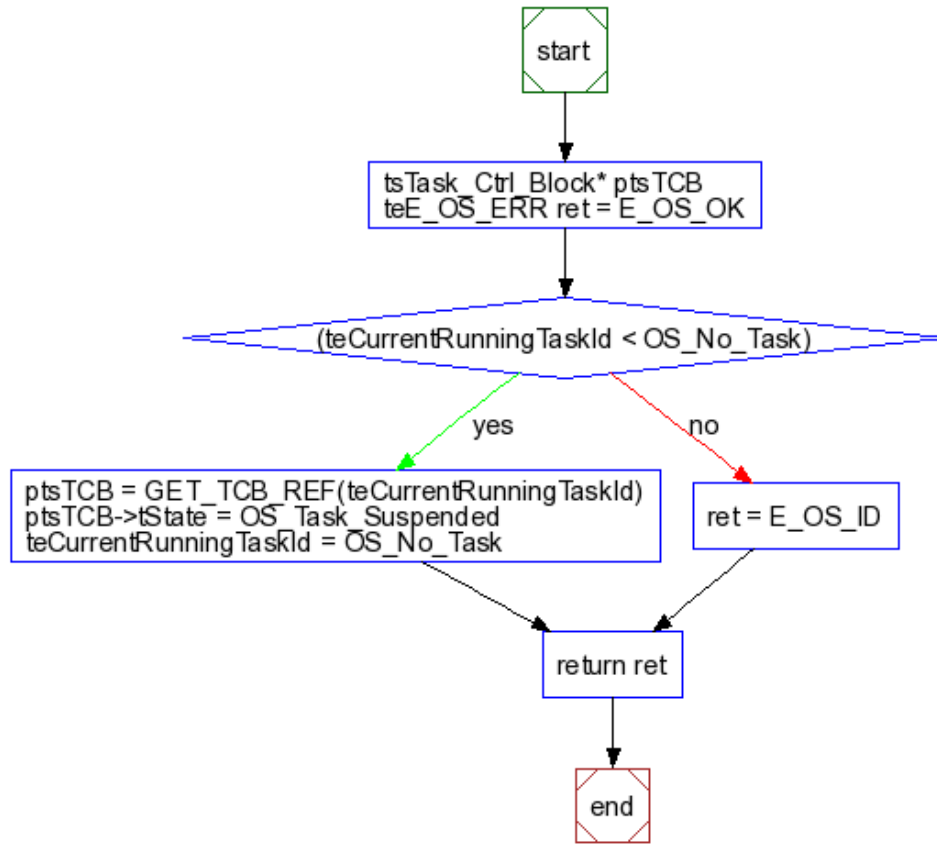


Figura 23. Diagrama de bloques de la función terminar.

3.7.5. Descripción de las operaciones de Background

La rutina de “Background” es una funcionalidad importante que cubre los requerimientos funcionales. En el siguiente diagrama se muestran las relaciones utilizadas para la ejecución la rutina del despachador.

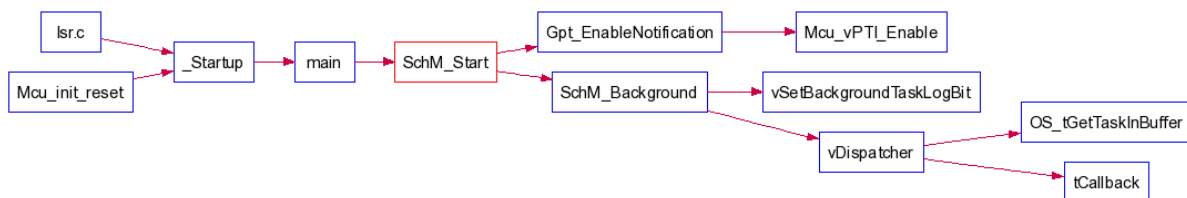


Figura 24. Secuencia de llamadas de función de la rutina de background.

Esto es, el primer paso es habilitar las notificaciones del tiempo base con el fin de llamar a la función de activación de tareas. Después de la ejecución de la rutina de interrupción, la planeación se ejecuta la rutina de la tarea de “Background” la cual ejecuta las llamadas a función de las tareas asociadas según su identificador que haya sido activado y puesto en espera. La descripción de esta rutina se muestra en el siguiente diagrama.

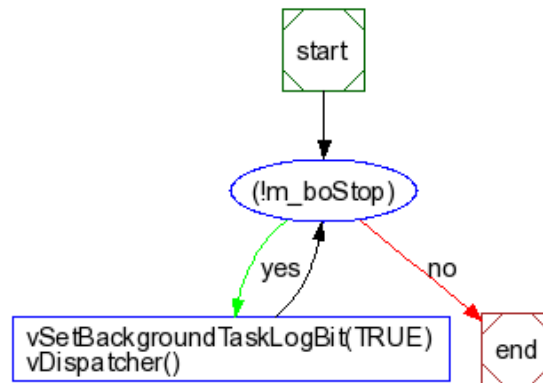


Figura 25. Diagrama a bloques de la rutina de background.

3.7.6. Servicio de Interrupción del SO

A continuación, se muestran las rutinas de cambio de contexto y servicios de interrupción utilizados para obtener un SO *preemptive*.

Cambio de contexto

```

#define SaveContext()                                     \
{                                                         \
    extern volatile void* currentTCB_Stack;             \
    extern void Os_vWrapperDispatcherFunction(void); \
    __asm( "ldaa 0x30" ); /* 0x30 = PPAGE */             \
    __asm( "psha" );                                     \
    __asm( "des" );                                     \
    __asm( "ldx currentTCB_Stack" );                   \
    __asm( "movw 0,SP,0,X" );                           \
    __asm( "movw 2,SP,2,X" );                           \
    __asm( "movw 4,SP,4,X" );                           \
    __asm( "movw 6,SP,6,X" );                           \
    __asm( "movw 8,SP,8,X" );                           \
    __asm( "movw 10,SP,10,X" );                         \
}
  
```

Restauración de contexto

```

#define RestoreContext()                                 \
{                                                         \
    extern volatile void* currentTCB_Stack;             \
    \
  
```

```

    __asm( "ldx currentTCB_Stack" );
    __asm( "movw 0,X,0,SP" );
    __asm( "movw 2,X,2,SP" );
    __asm( "movw 4,X,4,SP" );
    __asm( "movw 6,X,6,SP" );
    __asm( "movw 8,X,8,SP" );
    __asm( "movw 10,X,10,SP" );
    __asm( "ins" );
    __asm( "pula" );
        __asm( "staa 0x30" ); /* 0x30 = PPAGE */
        __asm( "rti" );
}

```

Rutina de interrupción del SO.

```

void OS_Entry_Mcu_vPIT_Channel0_Isr(void)
{
    SaveContext();
    INTERRUPT_DISABLE();
    Mcu_vHandle_PIT_Interrupt(PIT0);
    DispatcherService();
}

```

Ejemplo para la declaración de una tarea

```

TASK(vTask_4)
{
    vOS_Task_Work(OS_TASK_4_WORK_LOAD, PORTB_PB4_MASK);
    (void)TerminateTask();
    DispatcherService();
}

```

Ejemplo de activación de una tarea que active otra tarea.

```

TASK(vTask_3)
{
    vOS_Task_Work(OS_TASK_3_WORK_LOAD, PORTB_PB3_MASK);

    (void)ActivateTask(OS_Task_4);
    __asm( "swi" );

    vOS_Task_Work(OS_TASK_3_WORK_LOAD, PORTB_PB3_MASK);

    (void)TerminateTask();
    DispatcherService();
}

```


3.7.7. Operación

A continuación, se muestra el resultado operativo del SO implementado según las consideraciones de diseño establecidas en secciones anteriores. Tomando en cuenta que la tarea 3 realiza una actividad con una carga determinada, cuando se activa la tarea cuatro con mayor prioridad, está es ejecuta, interrumpiendo la tarea 3. Cuando la tarea 4 ha terminado entonces la tarea 3 continúa hasta fin de su rutina. Este comportamiento se puede visualizar en la gráfica de la figura 26.

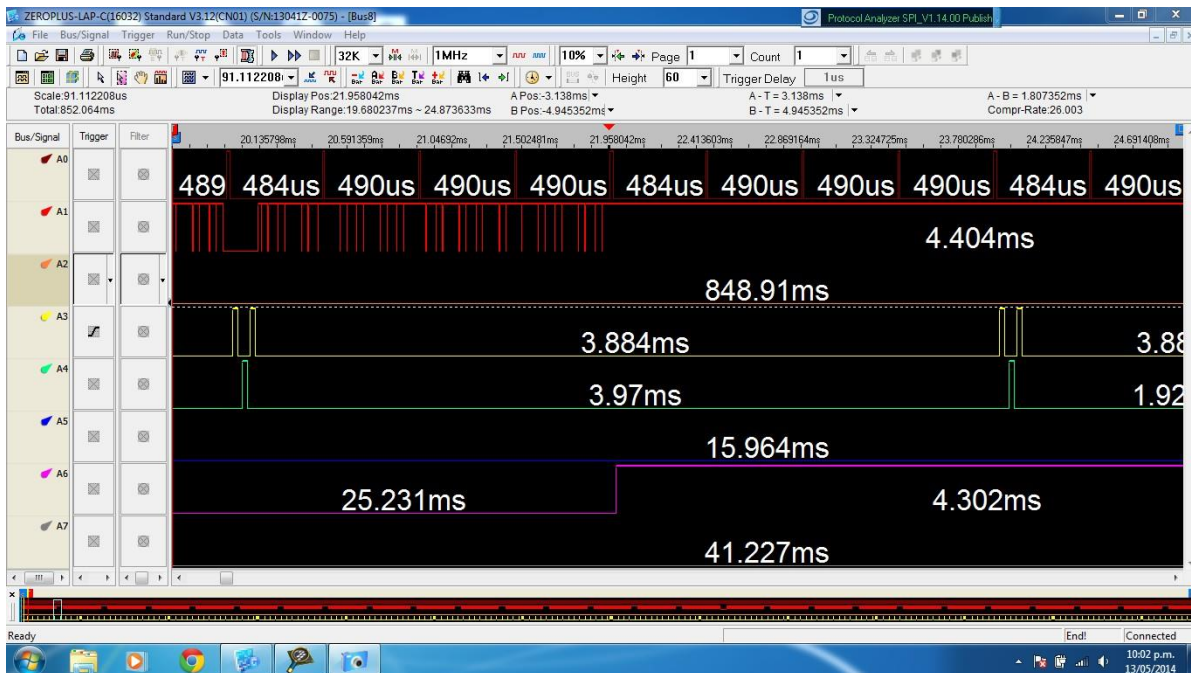


Figura 26. Diagrama de tiempo que muestra la ejecución de las tareas y el tiempo base.

Donde:

- Señal A1 representa la tarea de Background. Prioridad 0.
- Señal A3 representa la tarea 3. Prioridad 4.
- Señal A4 representa la tarea 4. Prioridad 5.
- Señal A5 representa la tarea 5 (No se muestra su actividad en este periodo). Prioridad 4.
- Señal A6 representa la tarea 6. Prioridad 3.

3.7.8. Configuración del manejador (Driver) de CAN

Fuente del Reloj y Velocidad de Transmisión:

El oscilador seleccionado para el módulo de CAN es el generador de pulsos proveniente del XTAL de la DEMOBoard 9S12. Este se describe en la Figura 27 y sus características en las ecuaciones subsecuentes.

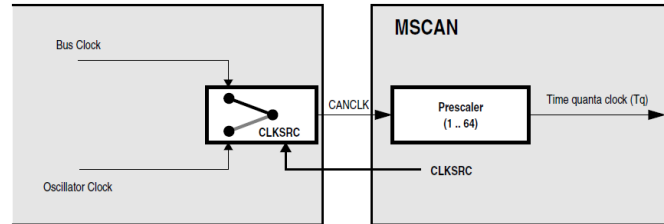


Figura 27. Fuente de reloj para el módulo de CAN.

$$\begin{aligned}
 \text{BaudRate} &= 500\text{kHz} && (1). \\
 \text{MSCAN_clock_source} &= \text{CANclk} = 8\text{MHz} && (2). \\
 \text{TSEG1} &= 5\text{quantas} && (3). \\
 \text{TSEG2} &= 2\text{quantas} && (4). \\
 \text{TSEG1} + \text{TSEG2} + \text{SYNCHSEG} &= 8\text{quantas} && (5). \\
 \text{Prescaler} &= \frac{\text{CANclk}}{(\text{BaudRate} * \text{timequantas})} = \frac{8\text{MHz}}{(500\text{kHz} * 8)} = 2 && (6).
 \end{aligned}$$

3.7.9. Circuito Convertidor

Para la construcción del dispositivo se seleccionaron los siguientes dispositivos:

Device	Quantity	Description	Footprint
D0	1	SiC, BYV410X-600	Generic\TO-220
D1	1	SiC, FEP16DT-E3	Texas Instruments\TO-220AB
D2, D3	2	SiC, DSEI30-06A	Generic\TO-247AC
C0	1	Capacitor, Vishay Cap MKP1848	Ultiboard\MKP1848_200
C1	1	CAP_ELECTROLIT, 56µF 10%	IPC-2221A/2222\CAPPR500-1000X1250
C2, C3	2	CAP_ELECTROLIT, 22µF 10%	IPC-2221A/2222\CAPPR500-1250X2500
Q1	1	MOS_3TEN, IRFP441	Generic\TO-247AC
L	1	Coupled Inductor ETD59	Ultiboard\ETD59

Tabla 6. Lista de materiales que componen el circuito convertidor.

El diseño de la placa de circuito es como sigue:

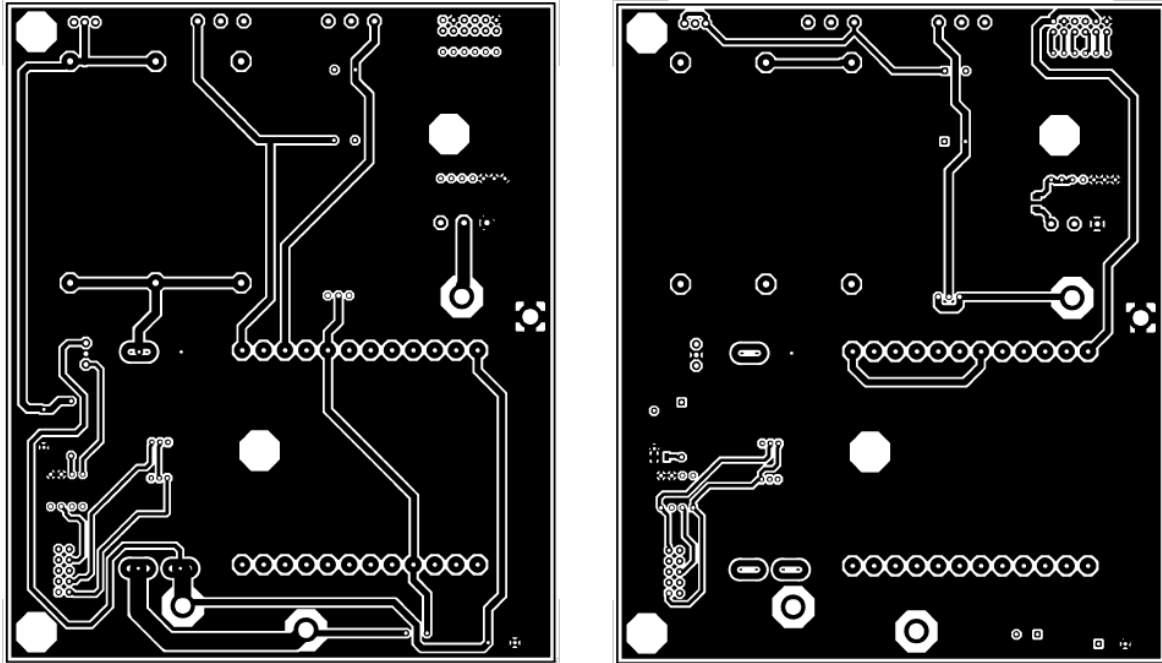


Figura 28. Plantilla PCB del circuito convertidor.

3.7.10. Sistema de Control

El siguiente diagrama muestra una representación del modelo del convertidor a través de sus ecuaciones de estado. Cada variable de estado se representa por un bloque independiente que tiene varias variables de entrada y una variable de salida, que como es de esperarse es el estado asociado al bloque.

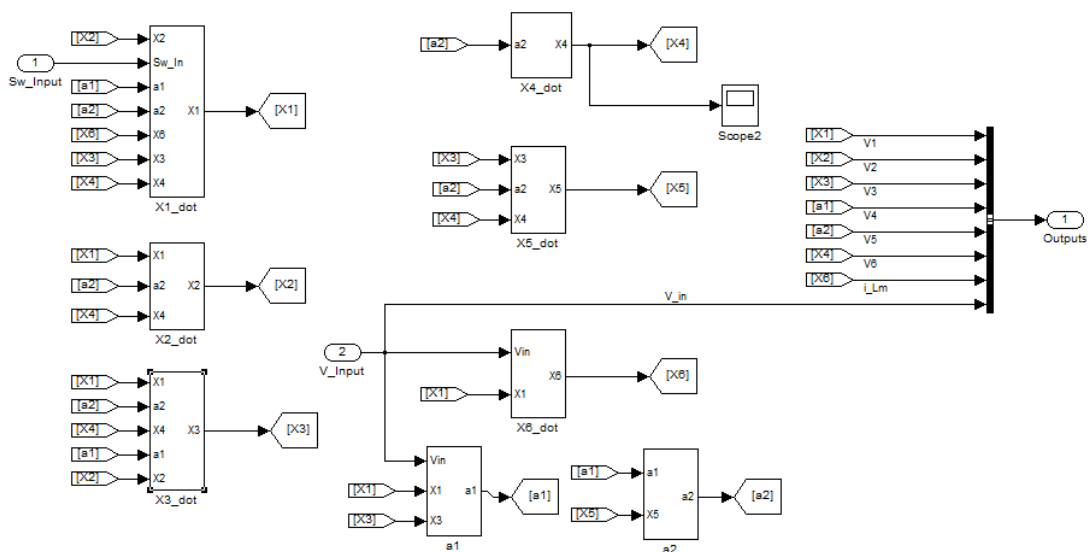


Figura 29. Diagrama del modelo del sistema en su representación a bloques.

La salida de la Figura 29 representa el vector de estados del sistema, el cual alimenta el diagrama de la Figura 30. Este último a su vez muestra la relación que existe en los flujos de corriente del circuito a través de los diodos y la influencia que tiene la capacitancia del Mosfet para el voltaje de entrada en el primario del inductor acoplado.

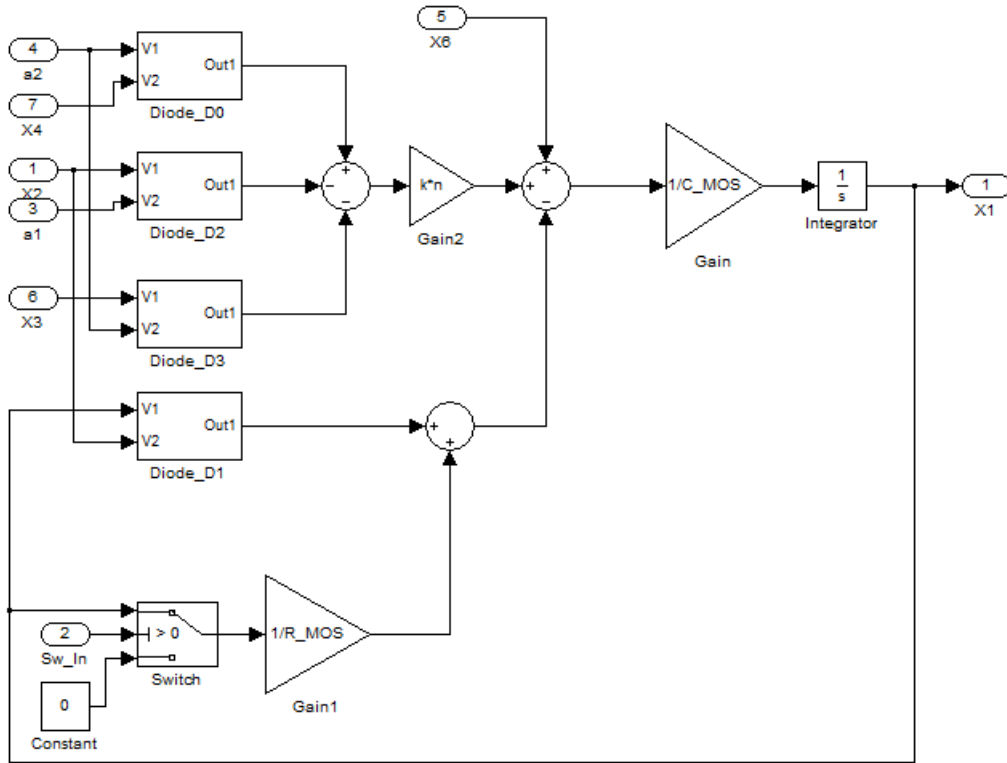


Figura 30. Representación en diagrama a bloques de la ecuación 2.6

Finalmente, como parte de los diagramas del modelo del sistema, se muestra el modelo simplificado de los diodos, el cual se aproxima por la función no lineal característica de un diodo para conducir corriente en polarización directa y estar en corte si la polarización es inversa. Al modelo se agregó un factor de compensación de temperatura V_T .

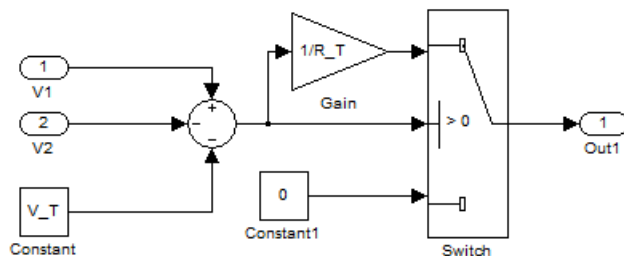


Figura 31. Representación en diagrama a bloques del modelo simplificado de operación del diodo.

En las gráficas de la Figura 32 se muestra la respuesta del sistema a una entrada tipo escalón, donde se puede ver el comportamiento del voltaje de salida, la corriente de magnetización, y porcentaje del ciclo de

duración de la señal de entrada que en este caso se trata de un cambio en el tiempo de duración de una señal PWM.

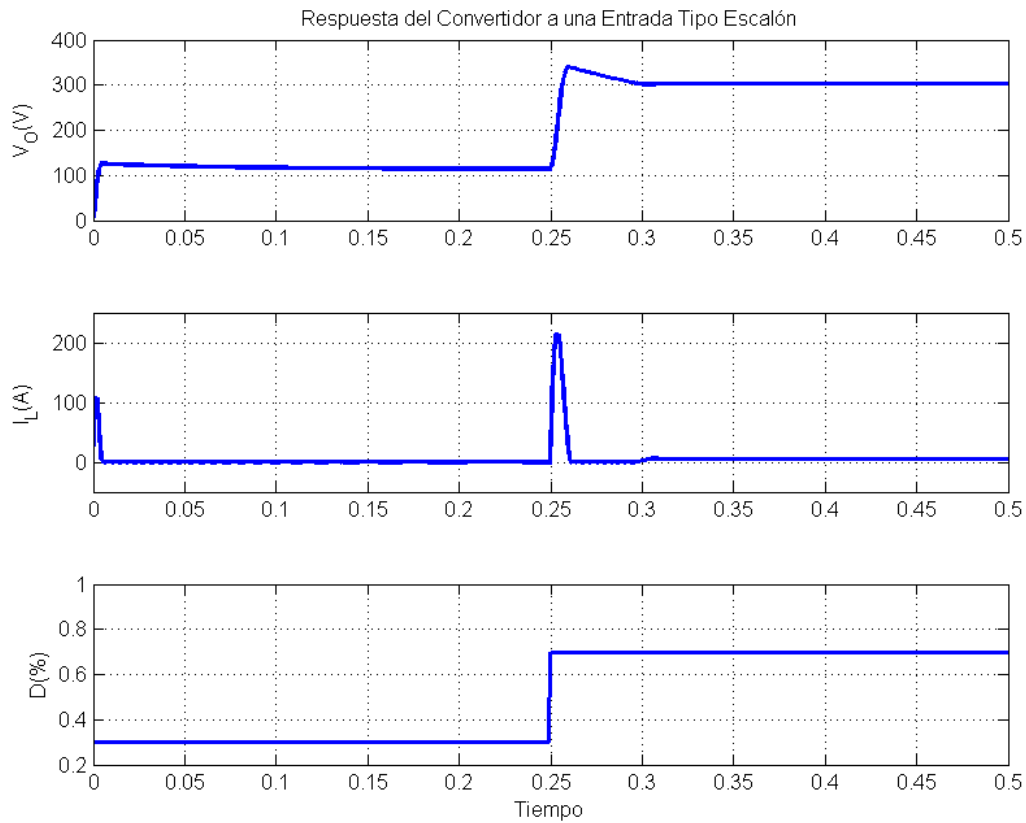


Figura 32. Respuesta del sistema a una entrada tipo escalón.

Finalmente, en la figura Figura 33, se muestra el resultado de aplicar el control por modos deslizantes al convertidor CD/CD. En los resultados se puede observar el fuerte impulso que recibe el convertidor con el fin de alcanzar el valor de referencia rápidamente, llevando la salida a voltajes por encima de 500 volts, generando un sobre impulso del 25% que se va reduciendo por un período de aproximadamente 50 ms hasta llegar a un valor muy cercano al de referencia.

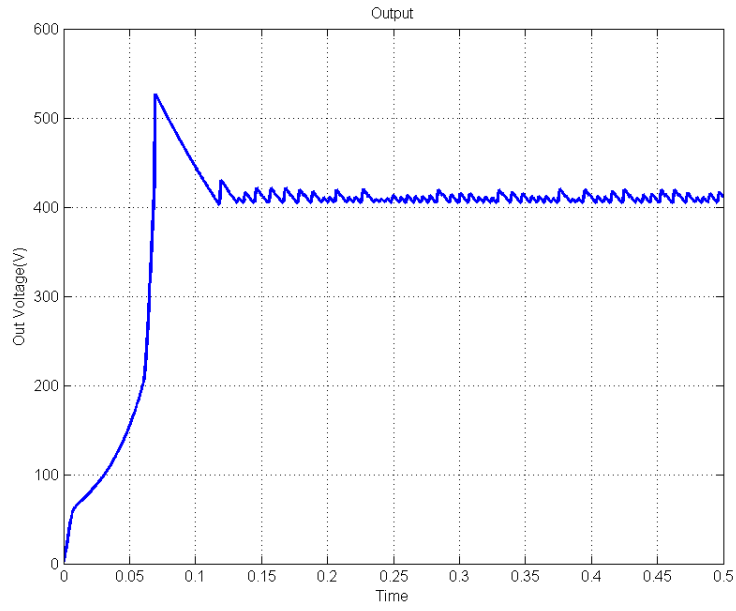


Figura 33. Respuesta lazo cerrado utilizando un control *SM* con retroalimentación de corriente.

4. RESULTADOS DE LA IMPLEMENTACIÓN EMBEBIDA

Resumen: En este capítulo se presentan las implementaciones desarrolladas en este trabajo para solucionar el problema planteado como objeto de estudio.

La fase de pruebas es de relevante importancia en el ciclo de desarrollo de un producto. De hecho, los procesos de prueba se hacen a lo largo del desarrollo del producto, que pueden ser revisiones de código (FTR), simulaciones, entre otras. Las pruebas permiten identificar problemas antes de llegar a fases más avanzadas como producción.

La fase de pruebas tiene su propio ciclo “V”, el cual se documenta en cada fase del desarrollo. Las pruebas se pueden realizar tanto a nivel de módulos (pruebas de código, eléctricas al PCB etc.) así como a nivel de sistema (probar el producto final como tal). La cantidad de pruebas a realizar a un determinado producto pueden variar según factores como el costo, sensibilidad, requerimientos.

En el caso de proyectos de gran envergadura, existen equipos dedicados a la realización de pruebas, preferentemente que no tengan conocimientos profundos acerca del proyecto. Con esto se espera que los problemas surjan y puedan ser resueltos sin obviar resultados. Es crítico que en esta etapa los problemas sean resueltos.

En el caso del convertidor de DC/DC, las pruebas que se deben realizar son de hardware y software, siendo las primeras las más complejas por el equipamiento requerido para hacer, por ejemplo, pruebas de compatibilidad electromagnética. Debido a ello se da principal atención a las pruebas de software como el medio asegurar el mejor funcionamiento posible del hardware.

Para la realización de las pruebas se utiliza el prototipo construido y algunas herramientas de instrumentación como osciloscopio multímetro y los sensores propios del convertidor. La comunicación se realiza entre un módulo BCM y el convertidor, a través del intercambio de mensajes CAN.

A continuación, se presentan los resultados de los casos de prueba.

4.1. Casos de Prueba del Planificador

La prueba del planificador se centra en determinar si las prioridades que fueron asignadas a las tareas son respetadas durante la ejecución de la aplicación. Por tratarse de un sistema que realiza tareas de control es imperante que se respete el periodo establecido para actualizar los valores de la salida, así como del muestreo de los valores de las variables de entrada.

El entorno de prueba es relativamente sencillo de establecer, pues solo se requiere de la ejecución del programa y las terminales del osciloscopio para observar el cambio de estado de las tareas de “Ready” a “Run”.

4.1.1. BPS_TC_1

Test Case	ID	Status
Tasks Execution	BPS_TC_1	Active
Requirements covered		
BPS_1.1 BPS_1.2 BPS_1.3		

BPS_1.4
 BPS_1.5
 BPS_1.6

Test Procedure

– From the tasks defined in the tasks table:
 › Turn a pin level ON at the entrance of a Task and turn the pin level OFF at the end of a task execution.

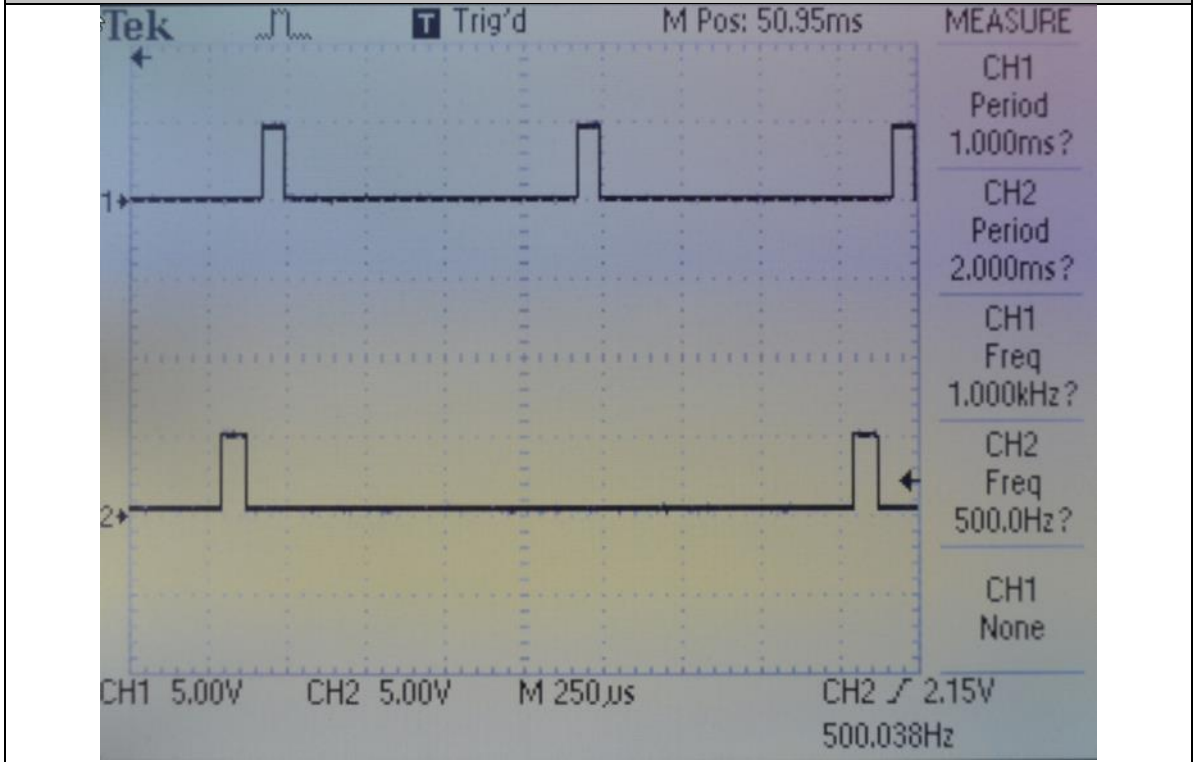
Expected Results

All task time execution, and task period execution should be measurable and values should match with the Task Table

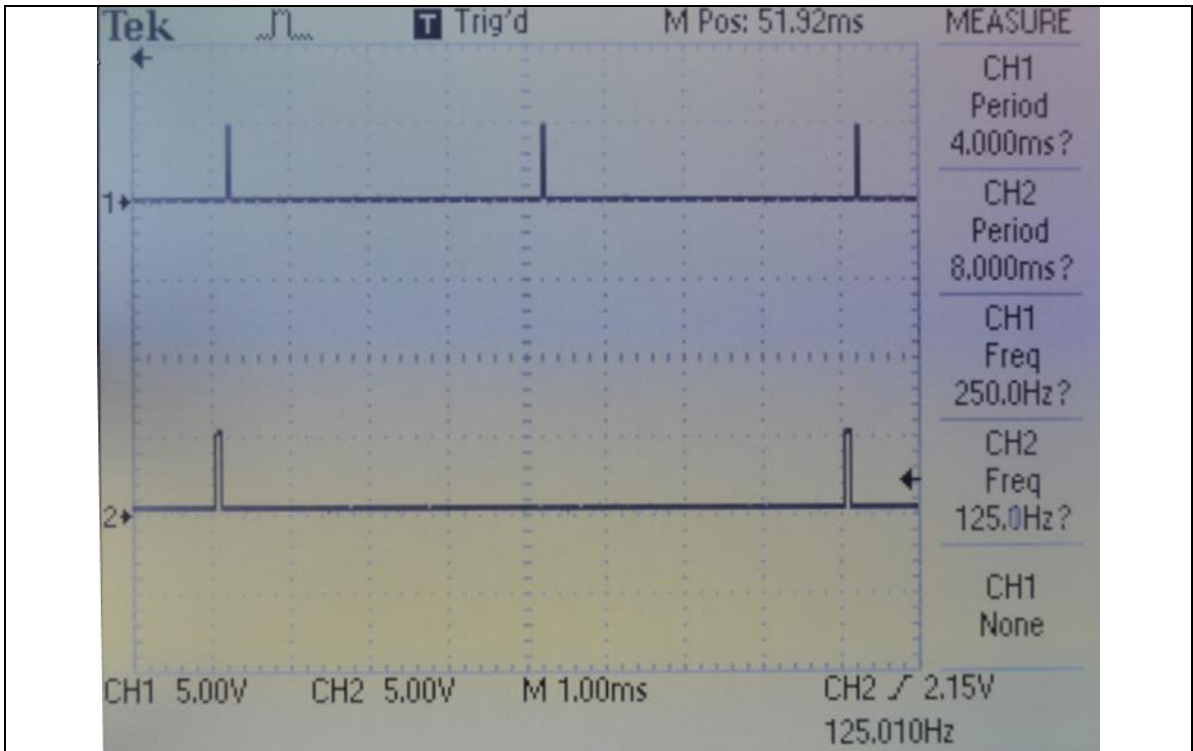
Actual Results	Test Results
----------------	--------------

<p>All tasks period measured and full filled the required periods. Screen shots attached in comments sections include all tasks period measurements.</p> <p>Note1: Task2 was also included (2ms period) since it is needed to meet following tasks periods, however this task can be removed from scheduler by setting its status to “Disabled” in the general configuration.</p>	<p>OK</p>
---	-----------

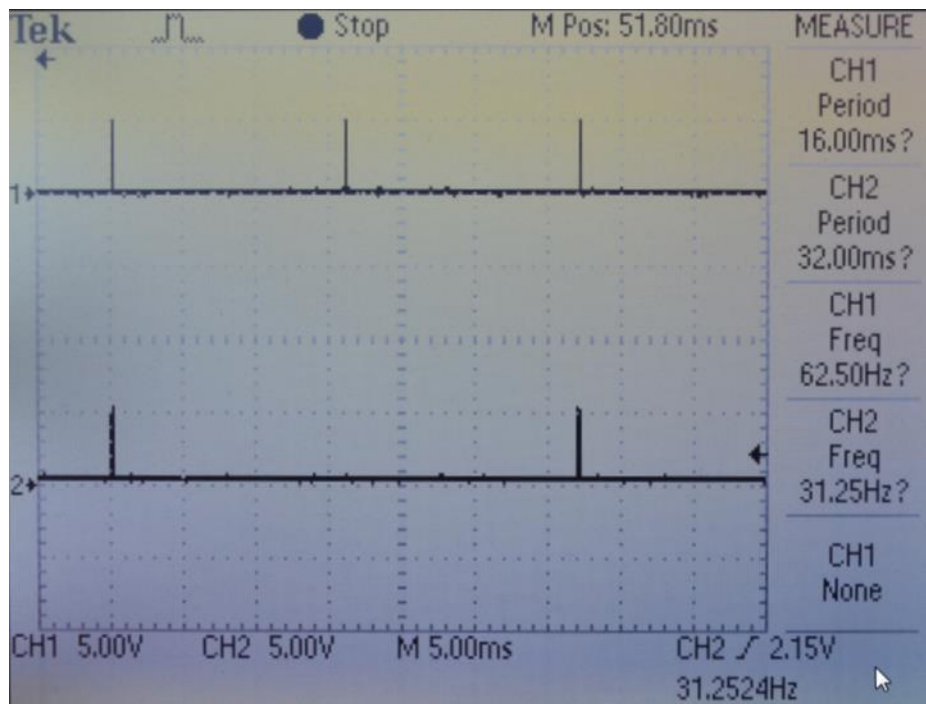
Comments



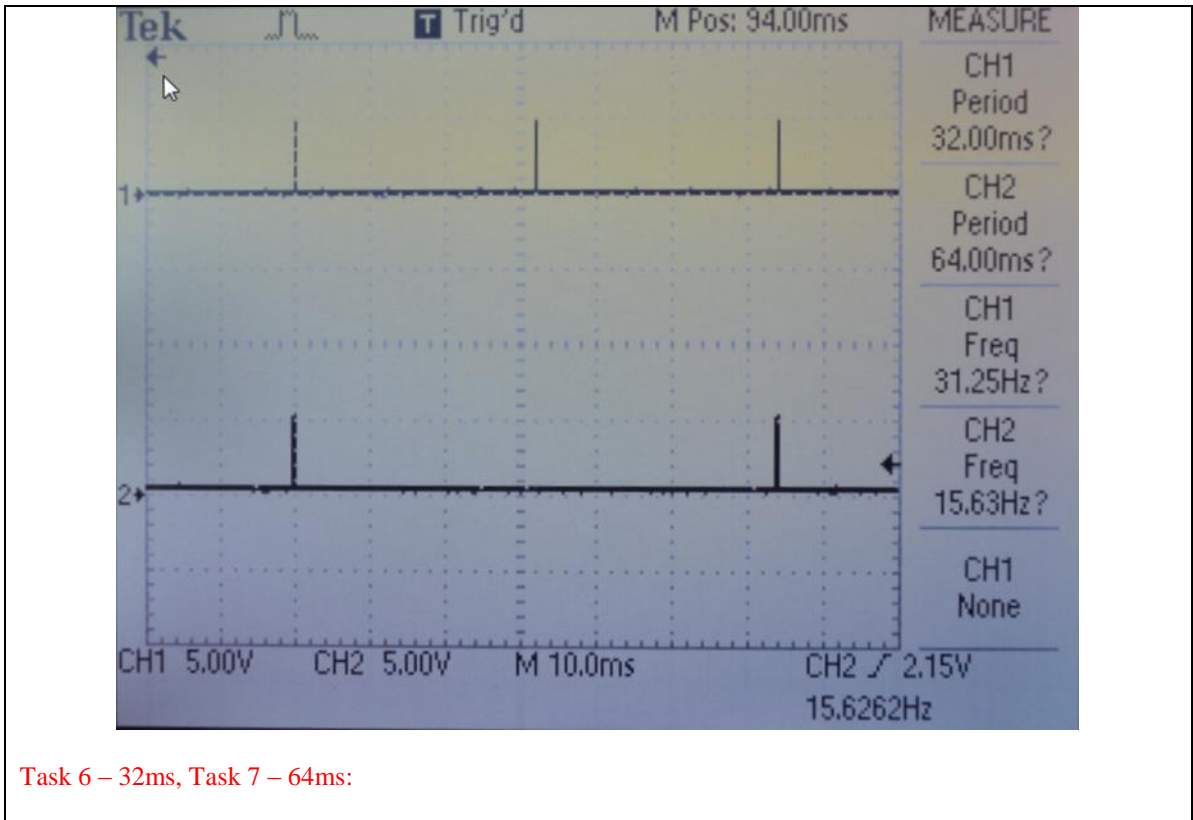
Task1 – 1ms & Task 2 – 2ms:



Task 3 – 4ms, Task 4 – 8ms:



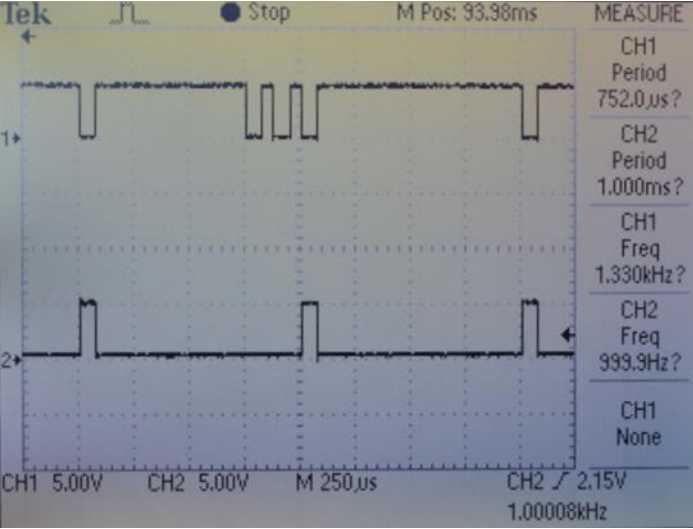
Task 5 – 16ms, Task 6 – 32ms:



4.1.2. BPS_TC_2

Para la estimación de la carga del CPU, se establece una relación entre el tiempo que dura ejecutándose la tarea de “Background” y todas las tareas registradas por el planificador. Para el caso del convertidor, el peor escenario ocurre cuando todas las tareas coinciden en el mínimo común múltiplo del periodo de las tareas. Un buen valor de referencia para el tipo de planificador utilizado es 62%, este valor proviene de un análisis de carga para sistemas con tareas periódicas que se conoce como “Rate Monotonic” en inglés. Por simple inspección se puede observar que el uso del CPU está por debajo del valor recomendado.

Test Case	ID	Status
Measure the current CPU load	BPS_TC_2	Active
Requirements covered		
BPS_1.7		
Test Procedure		
Turn a pin level ON before entering the Background Task and turn the pin level OFF at the end of the Background Task.		

Expected Results	
CPU Load should be inverse relative to the Background Task period.	
Actual Results	Test Results
Back ground task is active while no other Task is running. In the next picture it we can compare Background tasks vs. Task 1. However we also can see that certain “inactive” time slots in Background tasks, this slots are caused by other tasks execution.	OK
Comments	
	
Channel 1: Background Task, Channel 2: Task1	

4.1.3. BPS_TC_3

Para esta prueba se han utilizado los resultados del caso anterior, pero se revisa la robustez del planificador y su estimación del uso del CPU cambiando el periodo de prueba, esperando que la tarea de “Background” incremente la proporción del tiempo de uso del CPU.

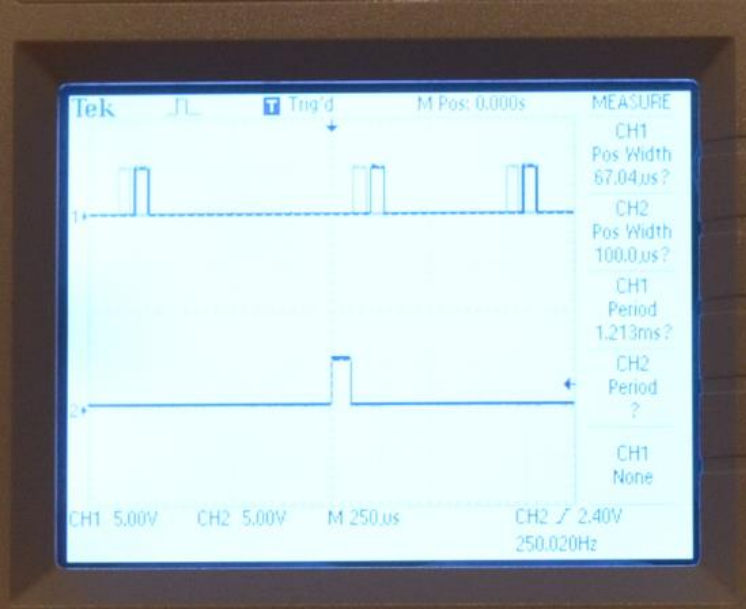
Test Case	ID	Status
Modify CPU load	BPS_TC_3	Active
Requirements covered		
BPS_1.7 BPS_1.8		

Test Procedure	
Modify the CPU Load by reducing workload to the tasks.	
Expected Results	
Back ground task execution should be now bigger, since normal task use less the CPU.	
Actual Results	Test Results
<p>When Tasks work load is reduced, it can be see how the inactive time slots in the Background tasks are reduced as well, which means the CPU load (CPU used by normal tasks) has also reduced.</p> <p>In the attached picture, it could be seen in Task1 that event its work load has been reduced, the task period remains in 1ms.</p>	OK
Comments	
<p>Channel 1: Background Task, Channel 2: Task1 – CPU Load reduced.</p>	

4.2. Casos de Prueba del Despachador

La siguiente prueba busca verificar el caso en que varias tareas que están configuradas con la misma prioridad son manejadas correctamente por el despachador, de tal suerte que se respete los periodos de cada una de ellas. Con esta prueba es posible verificar la correcta funcionalidad del manejo de la cola de tareas y el cambio de contexto que realiza el despachador.

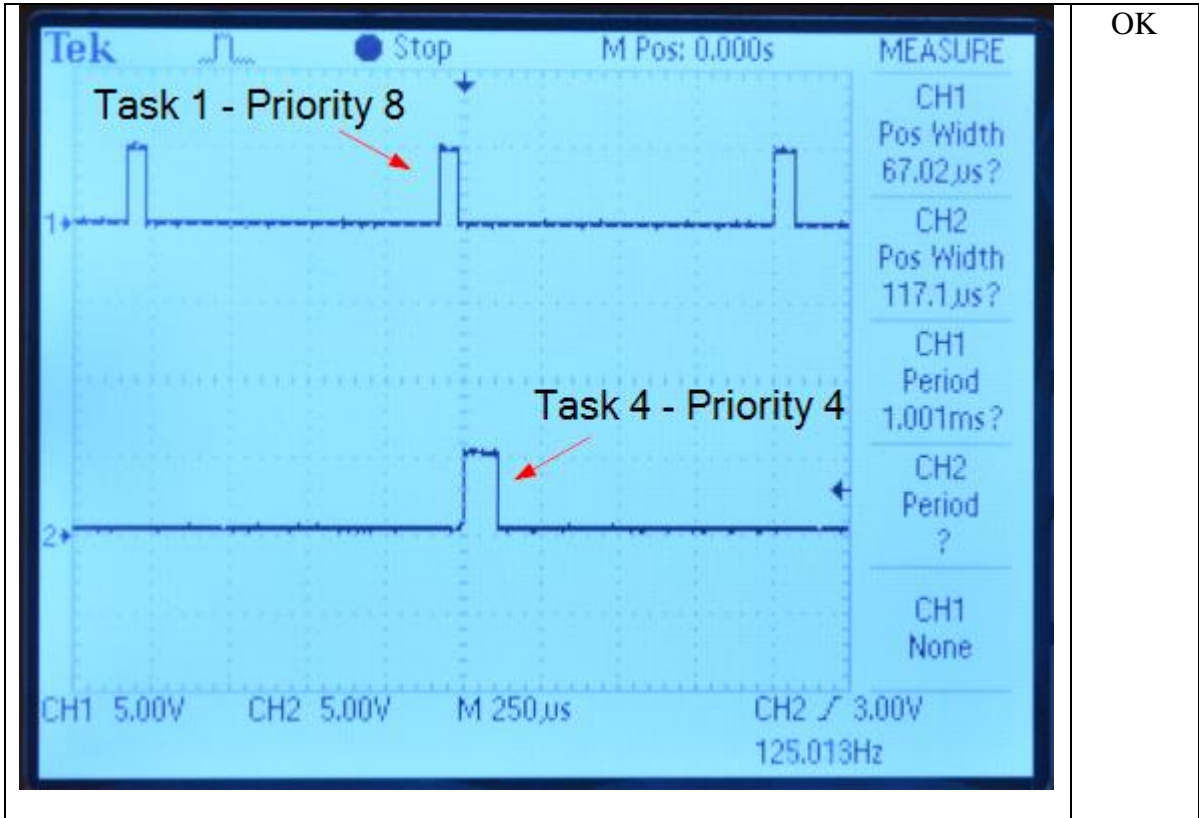
4.2.1. TMNP_TC_1

Test Case	ID	Status
Basic Operation Verification Tasks execution without Activation Overlapping.	TMNP_TC_1	Active
Requirements covered		
TMNP_3.1 TMNP_3.2 TMNP_4 TMNP_4.1 TMNP_4.2 TMNP_4.3		
Test Procedure		
Configure all tasks with same priority (4), and don't modify offset values so there's no tasks activation overlapping..		
Expected Results		
All tasks can be executed without overlapping at all. All task time execution, and task period execution should be. Also BackGroundTask time should be measurable.		
Actual Results		Test Results
		OK

4.2.2. TMNP_TC_2

Esta prueba consiste en probar que el despachador es capaz de realizar un cambio de contexto cuando una tarea de mayor prioridad requiere del uso del CPU a la vez que coincide con su tiempo de ejecución con una tarea de menor prioridad.

Test Case	ID	Status
High priority verification.	TMNP_TC_2	Active
Requirements covered		
TMNP_5		
TMNP_6		
TMNP_7		
TMNP_8		
Test Procedure		
Configure Task1 with: Offset = 0, and highest priority = 8.		
Expected Results		
Task1 activation will overlap with all other tasks, however Task1 should be executed first since has the highest priority.		
Actual Results		Test Results

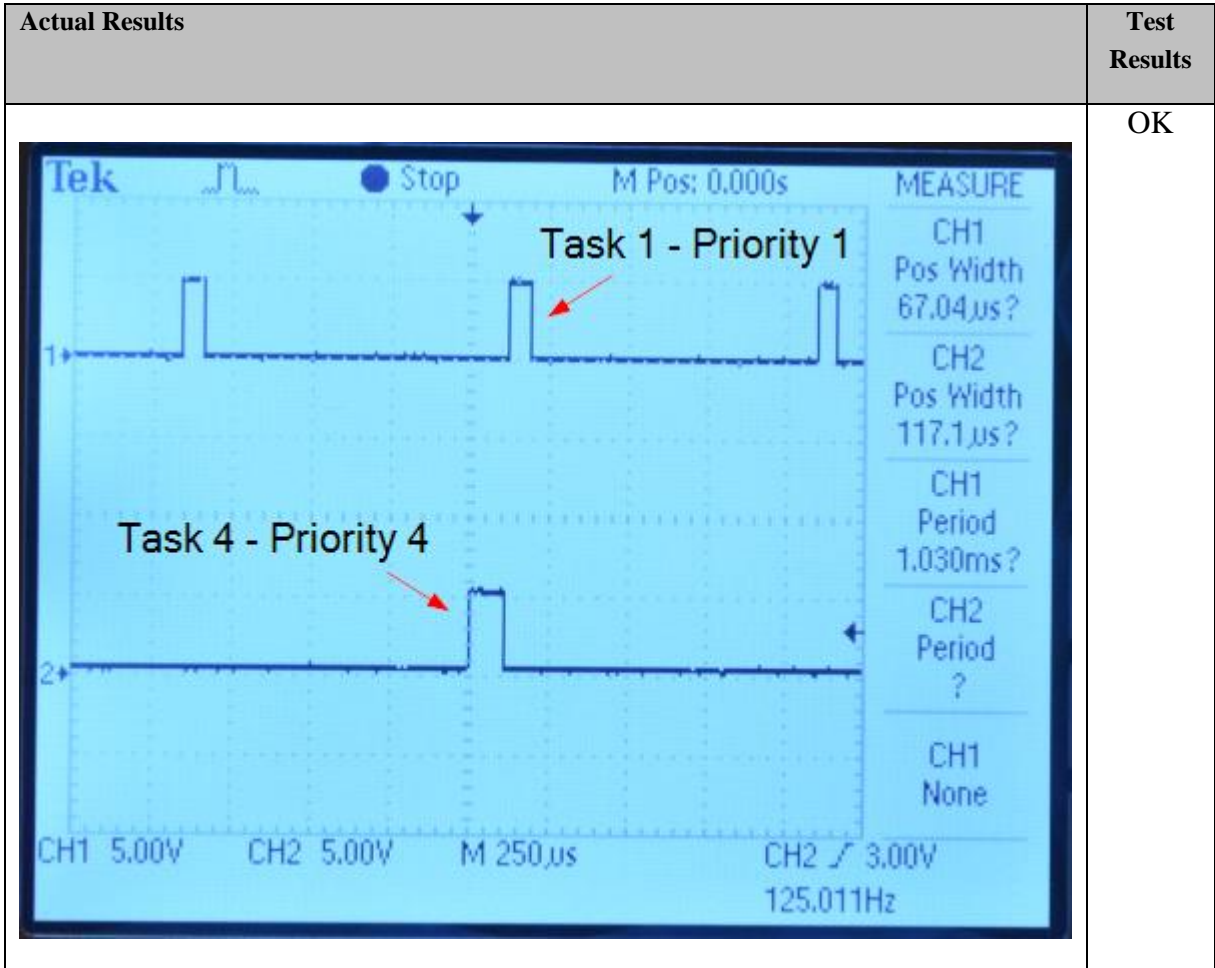


OK

4.2.3. TMNP_TC_3

A diferencia de la prueba anterior, aquí se busca probar la robustez del despachador verificando que una tarea de menor prioridad no interrumpa a una de mayor prioridad. Adicionalmente se verifica que una tarea de menor prioridad no interrumpa la ejecución de mayor prioridad incluso para decidir si es necesario un cambio de contexto.

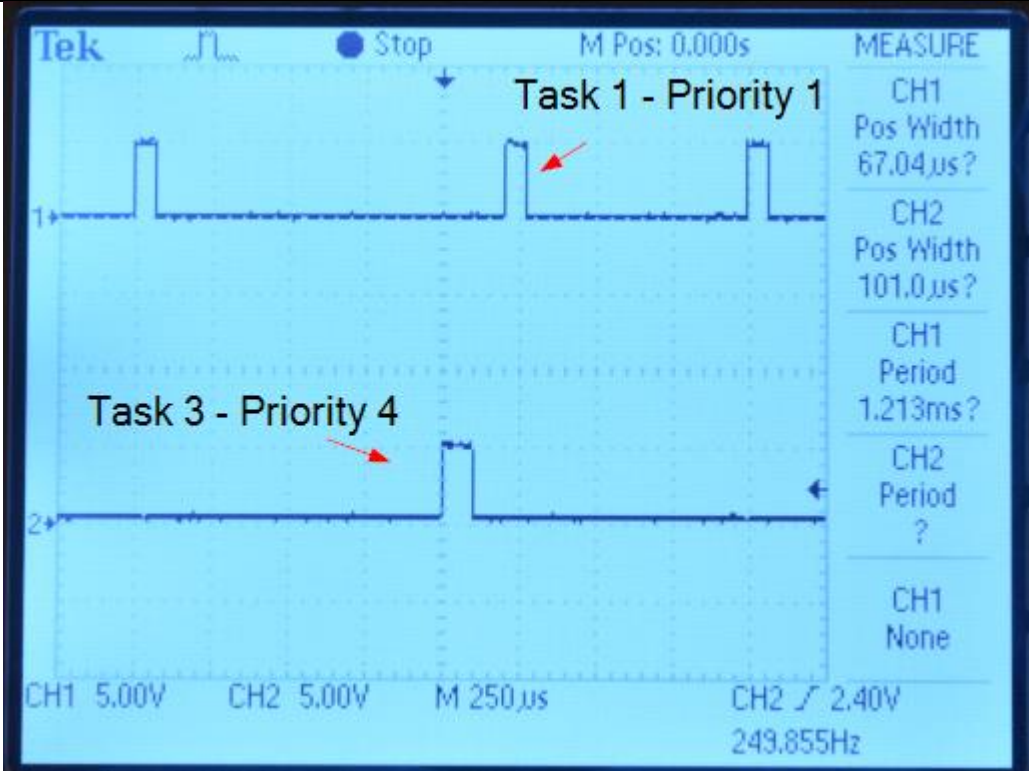
Test Case	ID	Status
Low priority verification.	TMNP_TC_3	Active
Requirements covered		
TMNP_5 TMNP_6 TMNP_7 TMNP_8		
Test Procedure		
Configure Task1 with: Offset = 0, and lowest priority = 1.		
Expected Results		
Task1 activation will overlap with all other tasks, however Task1 should be last executed since has the lowest priority.		

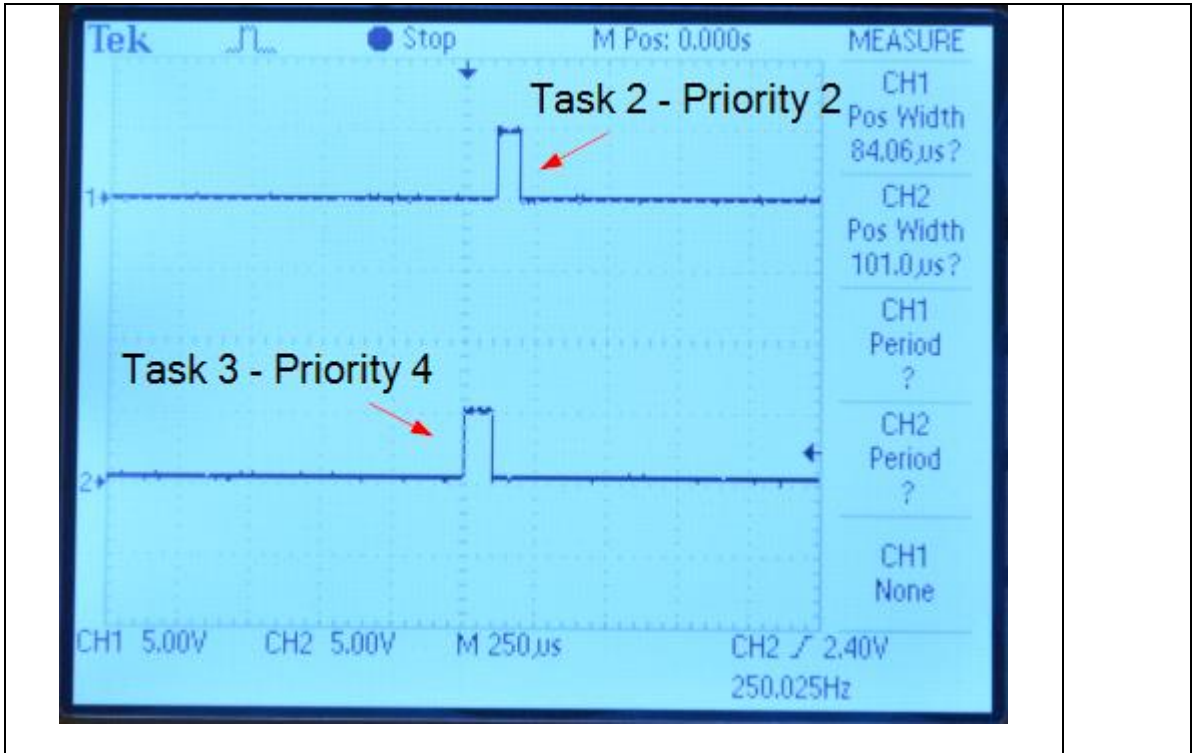


4.2.4. TMNP_TC_4

Esta prueba es realizada para verificar la robustez del despachador a través del incremento de tareas participantes en la ejecución del programa. Los objetivos de prueba son similares a las descritas previamente en esta misma sección, por lo que se sugiere consultarlas como referencia.

Test Case	ID	Status
Three tasks activation overlapping.	TMNP_TC_3	Active
Requirements covered		
TMNP_9 TMNP_10 TMNP_11 TMNP_12 TMNP_15 TMNP_15.1 TMNP_15.2 TMNP_15.3	TMNP_16 TMNP_17 TMNP_17.1 TMNP_17.2 TMNP_17.3 TMNP_17.4 TMNP_17.5	

Test Procedure	
Configure Task1 with: Offset = 0, and lowest priority = 1. Configure Task2 with: Offset = 0, and lower priority = 2.	
Expected Results	
Task1 and Task2 activations will overlap with all other tasks, however Task1 should be last executed since has the lowest priority, Task2 should be executed just before Task1, the other overlapped tasks should be executed first.	
Actual Results	Test Results
 <p>The screenshot shows two channels on a Tektronix oscilloscope. Channel 1 (CH1) is labeled 'Task 1 - Priority 1' and shows a periodic square wave with a period of 1.213ms and a pulse width of 67.04µs. Channel 2 (CH2) is labeled 'Task 3 - Priority 4' and shows a single pulse with a period of 101.0µs and a pulse width of 101.0µs. The oscilloscope settings are CH1 5.00V, CH2 5.00V, M 250µs, and CH2 2.40V. The frequency is 249.855Hz.</p>	OK



4.3. Casos de prueba cambio de contexto

4.3.1. PIV_TC1

En la siguiente prueba se pueden observar las interacciones entre varias tareas, el Temporizador del planificador y la tarea de “Background”. Para la realización de esta prueba se utilizó un analizador de estados lógicos, lo que permite observar un mayor número de tareas interactuando por un periodo de tiempo más largo que el de un osciloscopio tradicional.

Test Case	ID	Status
Basic Thread Preemption	PIV_TC_1	Active
Requirements covered		
PIV_1, PIV_2, PIV_3, PIV_4, PIV_5		
Test Procedure		
Enable all tasks to be executed under following priority conditions:		
Background task (Task 1): Priority 0 [Lowest priority]		
Scheduler task (Task 2): Priority 12 [Highest priority]		
All remaining Tasks set Priority 4.		

Task 6 assign a long period work and priority 3.

Expected Results

Background tasks, should be always preempted by any task.

Task 6 should be preempted by any other higher priority task tasks.

Scheduler task should never be preempted, and activation times should be stable.

When a task is preempted its context should be saved, and when the task execution is resumed the context should be restored.

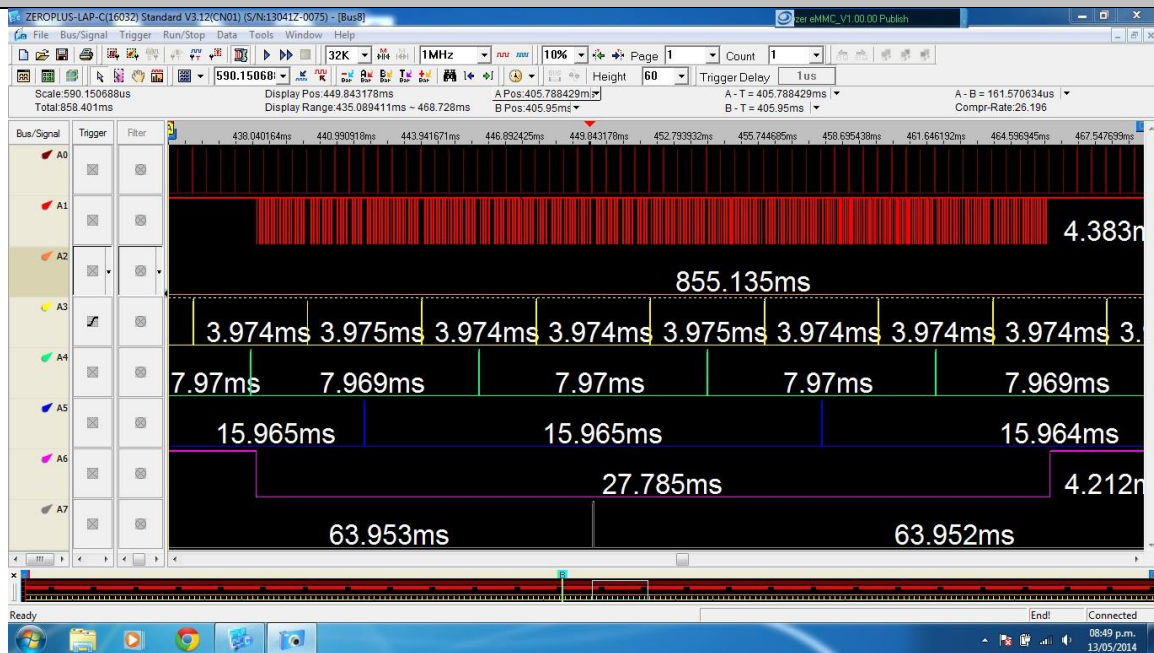
Actual Results

All results as expected. However it was noticed that with current implementation a running task can be preempted by other task with same priority, which is not completely correct, and can be solved at time the priority is evaluated in order to decide with current task should be preempted or not.

Test Results

OK

Comments



Graph 0: Connected to OsTick interruption. Shows ticks.

Graph1: Displays Background task.

Graph2: Connected to Scheduler task. It didn't updates the port output.

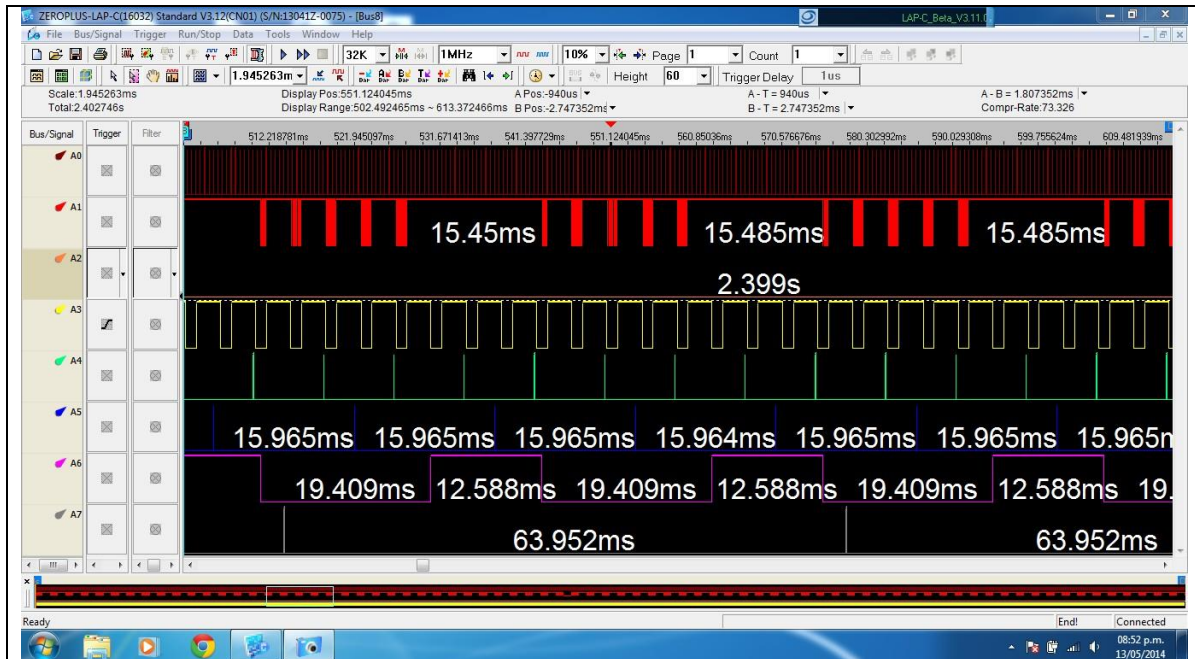
Graph3 to 6 connected to Task 3 to 6.

Here we can see Task 6 running time of 4.212ms, same time Background task is preempted. Also Task 6 gets preempted by other tasks Task3 and Task4.

4.3.2. PIV_TC_2

La siguiente prueba es realizada para verificar la robustez del algoritmo del cambio de contexto, para ello, partiendo de los resultados observados en la prueba anterior, se incrementa el tiempo de ejecución de una tarea significativamente, en este caso, 100 veces más que su valor en la prueba anterior. A pesar de los cambios, se espera que el sistema siga respetando los requisitos que se utilizaron de guía para el desarrollo del sistema, con la obvia diferencia que el consumo del CPU debe incrementar.

Test Case	ID	Status
Long running tasks preemption	PIV_TC_2	Active
Requirements covered		
PIV_1, PIV_2, PIV_3, PIV_4, PIV_5.		
Test Procedure		
Same procedure than prior test case 1. And increase the Running time of the task 3 for 100 times.		
Expected Results		
Since Task 6 is preempted by Task 3, the time needed to finish task 6 should reflect the time this task was preempted by task 3..		
Actual Results	Test Results	
Task 6 running time changed from 4.2 ms to 12.6 ms. Reflecting the time it was preempted by task 3.	OK	
Comments		



Now that Task 3 and Task 6 have long running times, it could be seen that the Background task execution time is reduced.

Save/Restore context is performed during the Tick ISR:

```
void OS_Entry_Mcu_vPIT_Channel0_Isr(void)
```

```
//ISR(Mcu_vPIT_Channel0_Isr)
```

```
{
```

```
    SaveContext();
```

```
    INTERRUPT_DISABLE();
```

```
    Mcu_vHandle_PIT_Interrupt(PIT0);
```

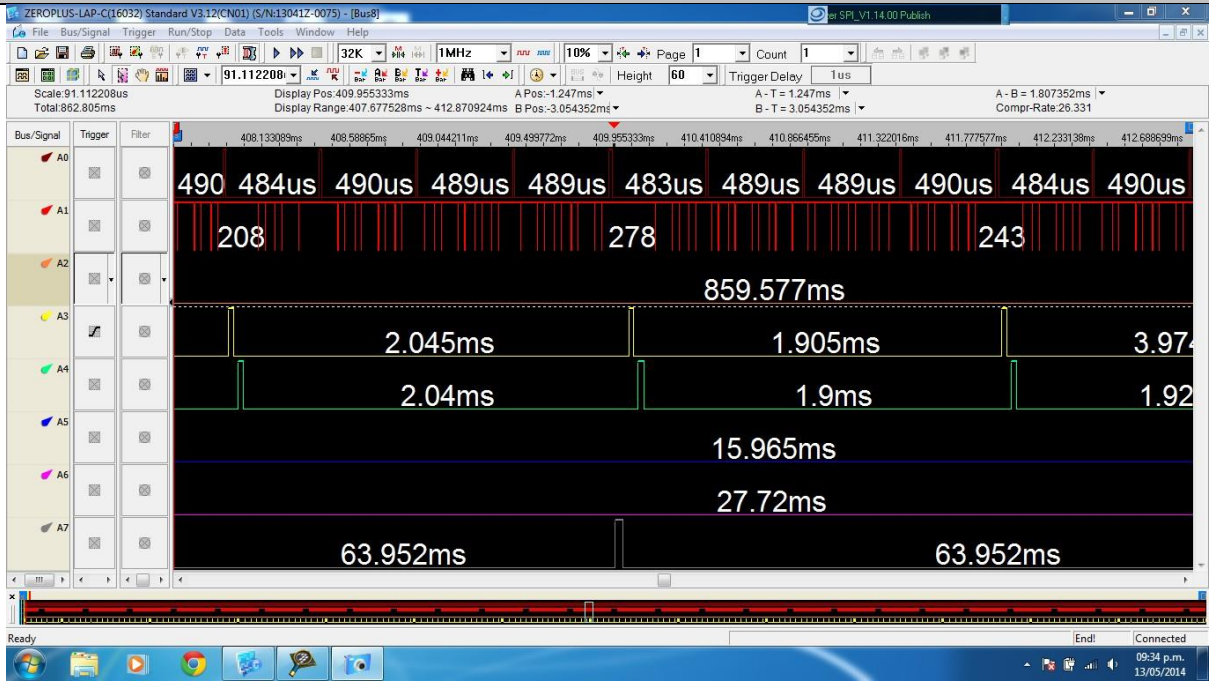
```
    DispatcherService();
```

```
}
```

4.3.3. PIV_TC_3

La siguiente prueba pretende verificar si la funcionalidad de cambio de contexto puede ser aplicada desde una instancia diferente al planificador, en este particular caso de uso se utiliza otra tarea. Esta funcionalidad podría parecer obvia en primera instancia, sin embargo, el manejo de eventos de una tarea a otra no es una acción fácil de realizar con la eficiencia requerida por un sistema que demanda tiempos de respuesta muy estrictos.

Test Case	ID	Status
Task activation from any other task	PIV_TC_3	Active
Requirements covered		
PIV_6, PIV_7, PIV_8, PIV_9, PIV_10.		
Test Procedure		
In test cases 1 and 2, the task activation was made by the Tick Interruption service routine. Now in this test case it should be also possible that One task can activate directly another task.		
Task 6 should activate task 3, and task 3 should activate task 4.		
Expected Results		
At the end of Task 6, Task 3 gets executed, but at the end of Task 3 Task 4 should be executed also.		
Actual Results		Test Results
Tasks execution performed as expected.		OK
Comments		



The task activation was set just after the current task finished its own work load.

TASK(vTask_3)

```

{
vOS_Task_Work(OS_TASK_3_WORK_LOAD, PORTB_PB3_MASK);

(void)ActivateTask(OS_Task_4);
__asm( "swi" );

(void)TerminateTask();
DispatcherService();
}

```

In order to be able to save/restore context using similar approach, Software Interrupts were used when one Task activates another Task.

```

void Mcu_vSWI_Isr(void)
{
SaveContext();

DispatcherService();
}

```

4.3.4. PIV_TC_4

Esta prueba es para verificar la robustez del sistema basado en el caso de prueba anterior. La expectativa es similar respecto a una tarea que activa a otra con la peculiaridad que para esta prueba se verifica que cuando una tarea de baja prioridad activa a una de mayor prioridad el cambio de contexto es inmediato, realizando la ejecución de la tarea de mayor prioridad para posteriormente dar lugar al cambio de contexto de la tarea inicial con el fin de terminar la ejecución en una “segunda fase” de operación.

Test Case	ID	Status
Task activation from any other task A	PIV_TC_4	Active
Requirements covered		

PIV_10_1, PIV_10_2, PIV_10_3

Test Procedure

Task 3 was configured to perform certain work load, then Activate Task 4 (higher priority), so task 4 gets executed. When task 4 finishes then Task 3 resumes and should perform a second work load.

Expected Results

Task 3 preempted by task 4, and task 3 work load split as expected..

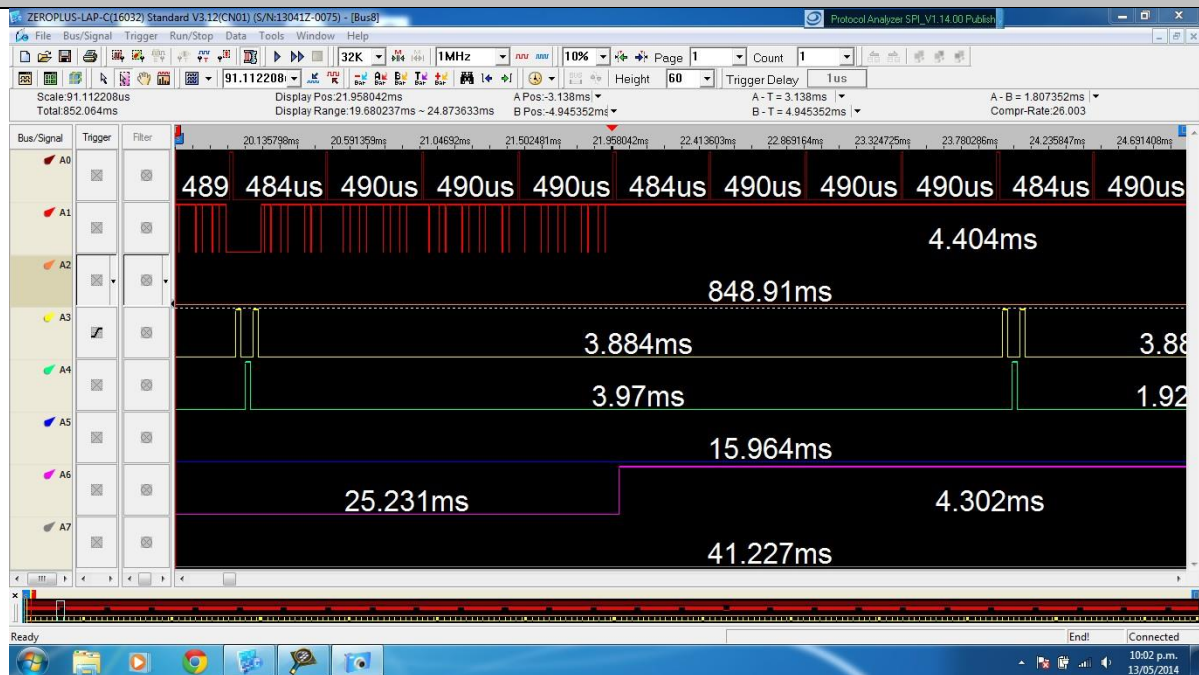
Actual Results

Tasks execution performed as expected.

Test Results

OK

Comments



TASK(vTask_3)

{

```
vOS_Task_Work(OS_TASK_3_WORK_LOAD, PORTB_PB3_MASK);
```

```
(void)ActivateTask(OS_Task_4);
```

```
__asm( "swi" );
```

```

vOS_Task_Work(OS_TASK_3_WORK_LOAD, PORTB_PB3_MASK);

(void)TerminateTask();

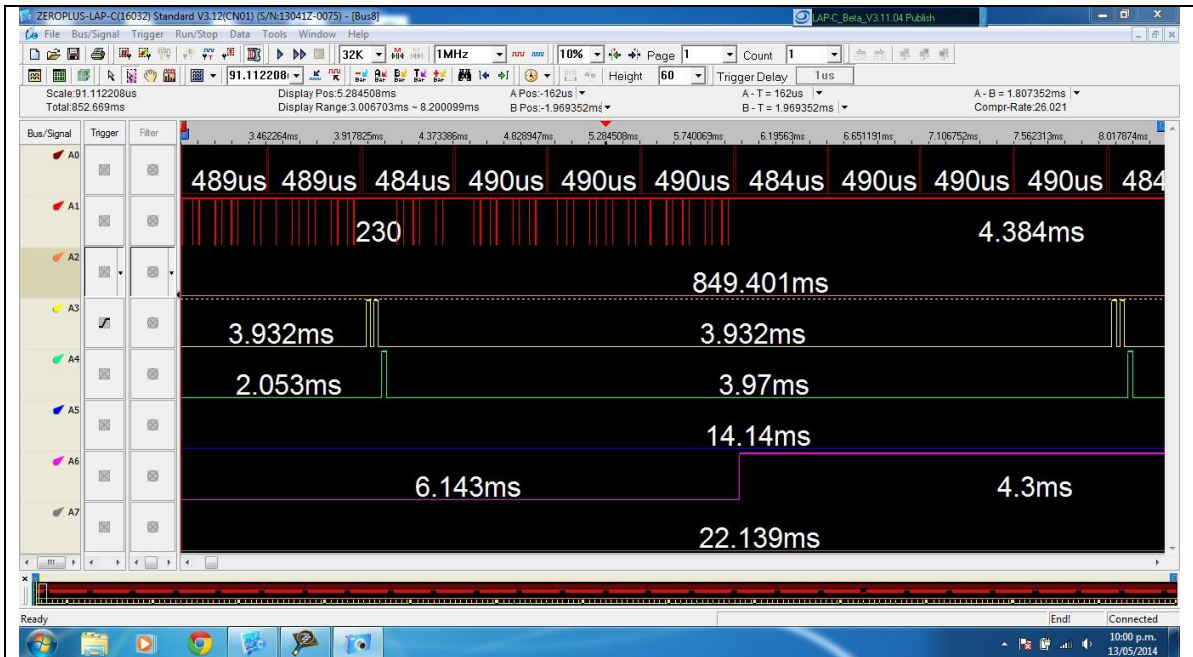
DispatcherService();
}

```

4.3.5. PIV_TC_5

Este caso de prueba es más simple que la anterior, puesto que no requiere de un cambio de contexto de la tarea, sin embargo, requiere de la participación del despachador para determinar la prioridad de la ejecución de la tarea que se está activando. Derivado de que la tarea recién activada es de menor prioridad, solo ocurre una breve interrupción en la ejecución de la tarea corriente, la cual al terminar da pie a que la tarea recién activada se pueda ejecutar como era esperado.

Test Case	ID	Status
Task activation from any other task A	PIV_TC_5	Active
Requirements covered		
PIV_10_1, PIV_10_2, PIV_10_3		
Test Procedure		
Task 3 was configured to perform certain work load, then performs a second workload and finally Activates Task 4 (higher priority), so task 4 gets executed.		
Expected Results		
Task 3 workload is finished, however before terminate Task 3, Task 4 is activated and executed (by the Dispatcher) and until Task 4 is terminated, context returns to Task 3 and gest terminated.		
Actual Results	Test Results	
Tasks execution performed as expected.	OK	
Comments		



```
TASK(vTask_3)
```

```
{
```

```
    vOS_Task_Work(OS_TASK_3_WORK_LOAD, PORTB_PB3_MASK);
```

```
    vOS_Task_Work(OS_TASK_3_WORK_LOAD, PORTB_PB3_MASK);
```

```
    (void)ActivateTask(OS_Task_4);
```

```
    __asm( "swi" );
```

```
    (void)TerminateTask();
```

```
    DispatcherService();
```

```
}
```

5. CONCLUSIONES

Resumen: En este capítulo se presentan las conclusiones del trabajo desarrollado y presentado.

Contar con fuentes de energía sustentable y amigable con el ambiente ha contribuido significativamente al interés del hombre por el desarrollo e investigación en el segmento de convertidores eléctricos de potencia. Por lo que las aportaciones tecnológicas en materia de conversión de energía son fundamentales para el uso y aprovechamiento de las llamadas energías limpias.

Prácticamente, los convertidores eléctricos pueden ser utilizados en cualquier proceso o actividad que realice el ser humano dónde requiera de una fuente de energía. En particular, se puede encontrar un creciente interés por aplicar este tipo de dispositivos en vehículos que no utilicen combustibles fósiles. Este comportamiento se puede observar, por ejemplo, en la reciente creación de una fórmula de autos de carreras totalmente eléctricos.

El combinar la implementación de un convertidor eléctrico con los requerimientos de una industria preocupada por la seguridad, como lo es la automotriz, conduce de forma natural a la obtención de productos de alto valor por su rango de aplicabilidad y confianza. Sin embargo, la obtención de tales productos, con semejantes cualidades, no pueden ser resultado de una sola disciplina sino de la combinación de varias de ellas. Tal es el caso del convertidor aquí presentado que reúne sinérgicamente disciplinas como la teoría de control y electricidad de potencia con la amalgama que provee la electrónica y la programación a través de los sistemas embebidos.

Como resultado del trabajo realizado se ha obtenido un prototipo de convertidor de CD/CD funcional de alta elevación capaz de suministrar energía hasta con una potencia de doscientos watts bajo un entorno variable con una salida estable. Para conseguir este comportamiento en el dispositivo fue necesario aplicar técnicas de control moderno, como la de control de corriente aplicando modos deslizantes, además de un sistema embebido con características de procesamiento en tiempo real duro, capaz de adherirse a un sistema global a través de la comunicación que ofrece el protocolo de CAN 2.0. Debido a la complejidad y diversidad de actividades que se deben realizar, se implementó el ciclo de desarrollo en V para administrar el ciclo de vida del desarrollo del convertidor.

Como herencia del trabajo realizado, se puede considerar que se tiene un prototipo alfa con las características necesarias para convertirse en prototipo beta. Donde las actividades del trabajo futuro se centrarían en la mejora del PCB de tal manera que haya un análisis de compatibilidad e interferencia electromagnética. También es necesario implementar una ley de control con mejor respuesta en estado estacionario, por ejemplo, un control integral de la superficie deslizante para sistemas de alto orden. Finalmente se debe considerar migrar el programa a un uC con soporte nativo para operaciones de punto fijo o flotante. Esto último con el fin de reducir el periodo de muestreo y respuesta del sistema. Aunque el carácter del proyecto no es con fines comerciales, ha sido altamente satisfactorio obtener un prototipo en aras de un futuro sustentable.

REFERENCIAS

- [1] Y. Yi-Ping Hsieh, "Novel High Step-Up DC–DC Converter for Distributed Generation System," vol. 60, no. 4, April 2013.
- [2] Siew-Chong Tan, Sliding Mode Control of Switching Power Converters, Florida: CRC Press, 2012.
- [3] T. J. Yang, "Transformer-less dc–dc converter with high voltage gain," IEEE Trans. Ind. Electron., vol. 56, no. 8, pp. 3144–3152,, Aug. 2009..
- [4] R. J. Wai, "High efficiency power conversion system for kilowatt-level stand-alone generation unit with low input voltage," vol. 55, no. 3702–3714, Oct. 2008.
- [5] A. Ioinovici, "Power Electronics and Energy Conversion Systems," in Power Electronics and Energy Conversion Systems, WILEY, 2013.
- [6] D. M. Mitchell, DC–DC Switching Regulator Analysis., New York: Mc-Graw Hill, 1998.
- [7] F. Bilalovic, "Buck converter regulator operating in the sliding mode," 1983.
- [8] R. Venkataramanan, "Sliding mode control of DC-to-DC converters," 1985.