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

REPORTE DE FORMACIÓN COMPLEMENTARIA

en área de Sistemas Embebidos

Trabajo recepcional que para obtener el título de

ESPECIALISTA EN SISTEMAS EMBEBIDOS

Presenta: Héctor Iván Chávez Rodríguez

Asesor:
Dr. Luis Rizo Domínguez

Tlaquepaque, Jalisco. Abril del 2018.

RECONOCIMIENTOS

Quiero agradecer al Consejo Nacional de Ciencia y Tecnología (CONCACYT) por su apoyo.

Gracias a la Especialidad en Sistemas Embebidos por brindarme la oportunidad de desarrollar las habilidades necesarias para mi realización profesional.

Quisiera expresar mi más sincero agradecimiento al Dr. Luis Rizo Domínguez, que fungió como mi asesor para poder lograr este trabajo.

Agradezco a mis Profesores Raúl Campos y Héctor Rivas por el material para el desarrollo de las prácticas reportadas en este trabajo.

Especialmente quiero agradecer al ITESO por el equipamiento y las instalaciones para lograr cada actividad desarrollada en este programa. Un agradecimiento especial a todos mis profesores por su paciencia, motivación y conocimiento que recibí de ellos.

TABLA DE CONTENIDOS

RECONOCIMIENTOS..... 2

TABLA DE CONTENIDOS..... 3
LISTA DE ACRONIMOS Y ABREVIATURAS..... 4

1. INTRODUCCIÓN..... 5

1.1. INTRODUCCIÓN..... 6

2. RESUMEN DE LOS PROYECTOS..... 7

2.1. SISTEMA DE CONTROL DE LUCES..... 8
2.1.1. INTRODUCCIÓN..... 8
2.1.2. ANTECEDENTES..... 8
2.1.3. DESARROLLO..... 9
2.1.4. ANALISIS DE RESULTADOS..... 11
2.1.5. CONCLUSIONES..... 11
2.2. CONTROL DE LA VELOCIDAD DE UN MOTOR BIPOLAR A PASOS..... 12
2.2.1. INTRODUCCIÓN..... 12
2.2.2. ANTECEDENTES..... 13
2.2.3. DESARROLLO..... 13
2.2.4. ANALISIS DE RESULTADOS..... 16
2.2.5. CONCLUSIONES..... 16

3. APENDICES..... 17

3.1. SISTEMA DE CONTROL DE LUCES REFERENCIAS..... 19
3.2. CONTROL DE LA VELOCIDAD DE UN MOTOR BIPOLAR A PASOS REFERENCIAS..... 24

4. TABLA DE DOCUMENTACIÓN..... 31

LISTA DE ACRONIMOS Y ABREVIATURAS

V-CYCLE	**GRAPHICAL REPRESENTATION OF A SYSTEMS DEVELOPMENT LIFECYCLE**
FMEA	Failure modes and effects analysis
QFD	QUALITY FUNCTION DEPLOYMENT
BOUNDARY DIAGRAM	Graphical illustration of the relationships between systems
FUNCTIONAL PROCESS DIAGRAM	Planning and management tool that visually describes the flow of work
FMEA	Step when preparing for a System or Subsystem FMEA
AUTOSAR	UTomotive Open System Architecture
STAKEHOLDER	A person, group or organization that has interest or concern in an organization
MICROCHIP	American manufacturer of microcontroller, memory and analog semiconductor
PIC	Family of microcontrollers made by Microchip Technology
PICOS18	Real time kernel based on the OSEK/VDXÔ standard for the automotive industry
RTOS	Operating system that has been developed for real-time
KERNEL	Computer program that is the core of a computer's operating system
SCHEDULER	Method that is used to distribute valuable computing resources
API	Application Programming Interface
PWM	Pulse-width modulation

1. INTRODUCCIÓN

Resumen: *Este capítulo presenta un resumen de este documento y los trabajos reportados.*

1.1. Introducción

El objetivo de este documento es describir el trabajo de formación complementaria realizado en el área de Sistemas Embebidos. Las materias de concentración y proyectos que se eligieron para este trabajo son los siguientes:

- **Ingeniería de Software para ambientes embebidos:**
  - Proyecto: Sistema de control de luces
- **Diseño de sistemas operativos para ambientes embebidos:**
  - Proyecto: Control de la velocidad de un motor bipolar a pasos

En primer lugar, se analiza y desarrolla un sistema de control de luces de un vehículo siguiendo la metodología de desarrollo V-cycle. Este proyecto empieza desde el análisis de requerimientos, pasando por la arquitectura y así sucesivamente por cada peldaño de esta metodología. Para este proyecto se crearon diversos documentos detallados para cada fase, tales como: FMEA, arquitectura basada en AUTOSAR, tabla de verdad, entre otros. El segundo proyecto consistió en la realización de un controlador digital para la regulación de la velocidad de un motor bipolar a pasos, por medio de la programación en lenguaje C de un Microcontrolador PIC de Microchip, usando PICOS18 como sistema operativo en tiempo real.

2. RESUMEN DE LOS PROYECTOS

Resumen: *En este capítulo se presenta un resumen de los trabajos y proyectos finales seleccionados.*

2.1. Sistema de control de luces

2.1.1. Introducción

El Método-V describe las actividades y los resultados que se producen durante el desarrollo del software. En este proyecto se realizó el análisis de la creación de un sistema de luces de un vehículo, pensando y creando todos los requerimientos de software, hardware, mecánicos y la unión de estas interfaces. Posteriormente viene la arquitectura de software basada en AUTOSAR, mostrando la relación que tiene cada módulo del software, pasando por la implementación del mismo. Posterior a esto, se valida la implementación en base a los requerimientos y arquitectura creada.

2.1.2. Antecedentes

El modelo en V es una variación del modelo en cascada que muestra cómo se relacionan las actividades de prueba con el análisis y el diseño. El modelo en cascada es muy rígida y estricta al no permitir regresar a sus anteriores fases de desarrollo con tal facilidad. Por lo tanto el modelo en V hace una mejor opción para proyectos grandes, siendo adoptada esta metodología por muchas industrias. Parte de las iteraciones y repeticiones de trabajo que están ocultas en el modelo en cascada. Mientras el foco del modelo en cascada se sitúa en los documentos y productos desarrollados, el modelo en V se centra en las actividades y la corrección.

2.1.3. Desarrollo

Requerimientos generales

Se realizaron diferentes documentos para demostrar el desarrollo del proyecto SISTEMA DE CONTROL DE LUCES:

- Requerimientos de sistema y software.
- House of quality.
- Boundary diagram.
- Functional process diagram.
- Software AUTOSAR architecture.
- Parameter Diagram
- Software test report
- FMEA

Requerimientos de sistema y software

Se creó un documento el cual muestra todos los requerimientos que se pensaron necesarios para la creación de este sistema.

House of quality

El QFD supone una metodología que permite sistematizar la información obtenida del usuario hasta llegar a definir las características de calidad del producto/servicio, adaptándolo al mercado.

Boundary diagram

Se realizó una ilustración gráfica de las relaciones entre los subsistemas, ensamblajes, subconjuntos y componentes dentro del objeto, así como las interfaces con los sistemas.

Functional process map

Es una herramienta de planificación y gestión que describe visualmente el flujo de trabajo. Los mapas de procesos muestran una serie de eventos que producen un resultado final.

Software AUTOSAR architecture

Se desarrolló una arquitectura de software basada en AUTOSAR, la cual es una arquitectura estandarizada desarrollada conjuntamente por fabricantes de automóviles, proveedores y desarrolladores de herramientas de software. El objetivo de esta colaboración es crear y establecer estándares abiertos para arquitecturas de componentes electrónicos en el sector automotriz.

Parameter Diagram

El Diagrama de Parámetros (P-Diagram) toma las entradas de un sistema / cliente y relaciona esas entradas con las salidas deseadas de un diseño que el ingeniero está creando, considerando también las influencias externas no controlables. Es una herramienta útil para generar ideas y documentar señales de entrada, factores de ruido, factores de control, estados de error y respuesta ideal.

Software test report

El software test report sirve para comunicar los resultados de los esfuerzos de prueba a las partes interesadas (por lo general, el equipo de desarrollo, incluidos los gerentes de proyectos, ingenieros, producción y qa). A continuación se muestra el test report de software del Sistema de control de luces.

FAILURE MODE EFFECTS ANALYSIS (FMEA)

El Modo de falla y análisis de efectos, es un enfoque estructurado para descubrir fallas potenciales que pueden existir dentro del diseño de un producto o proceso. Los modos de falla son las formas en que un proceso puede fallar. Los efectos son las formas en que estas fallas pueden generar desperdicio, defectos o resultados perjudiciales para el cliente. El modo de falla y análisis de efectos está diseñado para identificar, priorizar y limitar estos modos de falla.

2.1.4. Análisis de Resultados

En los resultados obtenidos, se puede mostrar el presente trabajo como un dibujo en el cual se necesita replantear algunos requerimientos tanto de sistema como de software y también de la parte de la arquitectura para poder decir que es una versión final del Sistema de control de luces.

2.1.5. Conclusiones

El presente proyecto permitió familiarizarnos y lograr un entendimiento de los beneficios que brinda la metodología de V-cycle. De igual manera, nos permitió entender y enfrentarnos a problemas en cada peldaño de esta metodología.

Este proyecto nos muestra que es necesario invertir más tiempo en el análisis de requerimientos y arquitectura para encontrar el menor número defectos en fases posteriores. También mejora la comunicación entre los Stakeholders y en definitiva, se trata de un modelo más robusto y completo que el Modelo de Cascada.

Finalmente, nos dio un claro entendimiento de cada uno de los roles de las diferentes disciplinas (eléctricos, mecánicos, software) y las interfaces que deben de proveer para poder crear un proyecto grande.

2.2. Control de la velocidad de un motor bipolar a pasos

2.2.1. Introducción

El trabajo consistió en la realización de un controlador digital para la regulación de la velocidad de un motor bipolar a pasos, por medio de la programación en lenguaje C de un Microcontrolador PIC de Microchip usando PICOS18 como sistema operativo en tiempo real. Para el logro del objetivo propuesto, se estudió el principio de funcionamiento de diversos driver controladores de motores, así como el de un motor a pasos, posteriormente se hizo uso de las APIS que nos ofrece el kernel de PICOS18 para poder realizar la practica mencionada.

2.2.2. Antecedentes

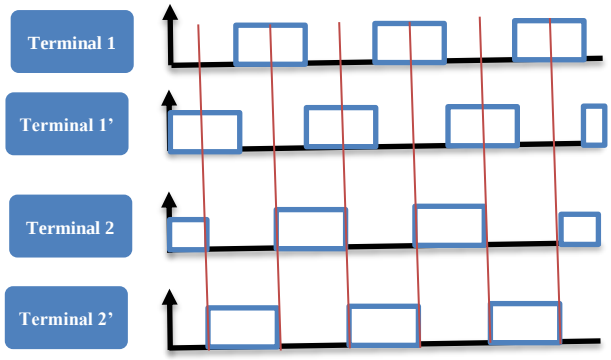
En la actualidad existen numerosas aplicaciones tanto industriales como académicas en las que se exige un control preciso y exacto de los actuadores implementados en tareas de transmisión de movimiento, posicionamiento y control de velocidad. Existen numerosas tarjetas de control para este tipo de motores, sin embargo, el costo y programación de las mismas en ocasiones complican la aplicación dentro de los proyectos donde se incorporan microcontroladores. La aplicación de un controlador que permita la variación de las señales al motor con frecuencias de alrededor de 625 Hz., y a la vez realizar diferentes tareas como es la comunicación y control de otros elementos es de gran importancia.

2.2.3. Desarrollo

Materiales, métodos y desarrollo

El proyecto consintió en la realización del control de la velocidad de un motor bipolar a pasos, en el cual por medio de por dos botones se hizo el incremento o decremento de la velocidad. Para esto se hizo uso de las interrupciones externas RB1 y RB2, para lo cual ambas se configuraron como interrupciones externas en el archivo main.c dentro de la función init.

Posteriormente se crearon por medio de la API SetRelAlarm los estados necesarios para crear 4 PWM que serán aplicados a cada terminal que se aplicara al motor como se muestra en la siguiente figura.



Secuencia de señal de excitación del motor a paso bipolar.

Paso	L1	L1'	L2	L2'
1	-	+	+	-
2	-	+	-	+
3	+	-	-	+
4	+	-	+	-

Secuencia de excitación de las bobinas del motor a paso bipolar.

El código utilizado para generar las señales presentadas se muestra a continuación:

```
TASK(TASK0)
{
    TRISDbits.TRISD2 = 0;
    TRISDbits.TRISD3 = 0;
    TRISDbits.TRISD4 = 0;
    TRISDbits.TRISD5 = 0;

    while(1)
    {
        SetRelAlarm(ALARM_TSK0, 1000, 30);
        WaitEvent(ALARM_EVENT);
        ClearEvent(ALARM_EVENT);

        counter++;

        if ((counter == duty) || (counter == period))
        {
            if(step == 1)
            {
                LATDbits.LATD2 = 1;
                LATDbits.LATD3 = 0;
                LATDbits.LATD4 = 0;
                LATDbits.LATD5 = 1;
                step ++;
            }
            else if(step == 2)
            {
                LATDbits.LATD2 = 0;
                LATDbits.LATD3 = 1;
                LATDbits.LATD4 = 0;
                LATDbits.LATD5 = 1;
                step ++;
             }
              else if (step == 3)
              {
                 LATDbits.LATD2 = 0;
                 LATDbits.LATD3 = 1;
                 LATDbits.LATD4 = 1;
                 LATDbits.LATD5 = 0;
                 step ++;
              }
              else if (step == 4)
              {
                 LATDbits.LATD2 = 1;
                 LATDbits.LATD3 = 0;
                 LATDbits.LATD4 = 1;
                 LATDbits.LATD5 = 0;
                 step ++;
              }
              else if (step == 5)
              {
                 step = 1;
              }
             if (counter == period)
              {
                  counter = 0;
              }
         }
     }
}
```

A continuación se muestra el código encargado del control de la velocidad del motor:

```
void MyOwnISR(void)
{
    //configure buttons
    WPUB = 0b00011111;
    INTCON2bits.RBPU = 0;

    if (INTCON3bits.INT1IF)
    {
          LATDbits.LATD0 = ~LATDbits.LATD0;
            if (duty >= 0)
            {
                duty = duty - 2;
            }

    INTCON3bits.INT1IF = 0;
    }

    if (INTCON3bits.INT2IF)
    {
          LATDbits.LATD7 = ~LATDbits.LATD7;
          if (duty <= period)
          {
              duty = duty + 2;
          }

       INTCON3bits.INT2IF = 0;
    }
}
```

Figura 8.-Codigo implementado dentro de la función de interrupción.

Al ser presionados los botones encargados de generar la interrupción, validarán el estado de la variable duty y period. Al presionar el botón RB2, el valor de duty será incrementado y esta aumentara el duty cycle por lo que aumentara su velocidad, y lo mismo procedimiento pasara al ser presionado el botón RB1 con la diferencia se decrementará el valor de duty y por consiguiente su velocidad.

Posteriormente, se realizó la conexión entre el microcontrolador a un puente H de transistores el cual permite amplificar la corriente que se le aplicara al motor, el diagrama se muestra a continuación:

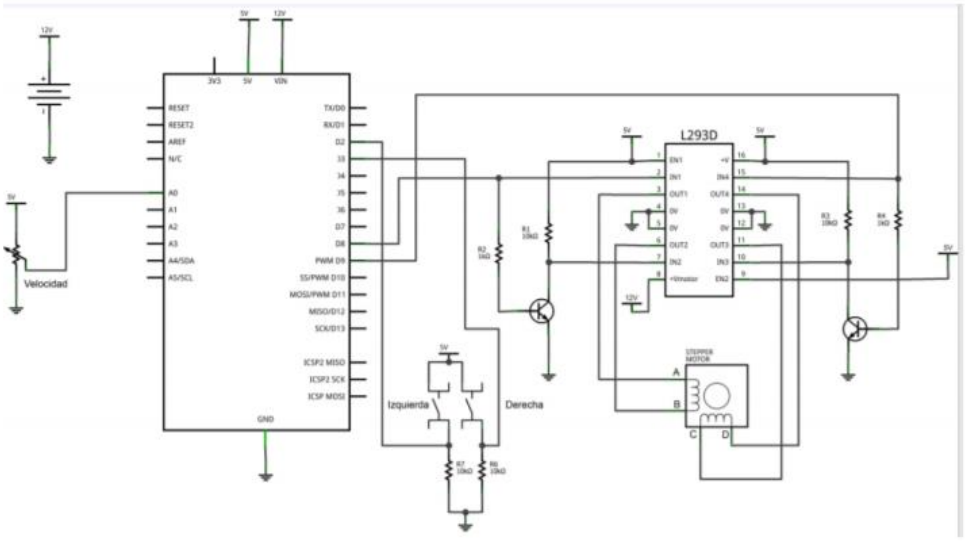


Figura 9.-Diagrama de control motor paso a paso.

Diymakers.es, "MOVER MOTORES PASO A PASO CON ARDUINO"(2016), http://diymakers.es/mover-motores-paso-paso-con-arduino/.

2.2.4. Análisis de Resultados

La implementación de este proyecto, llevó a cabo funciones dentro de tareas dentro de un tiempo especificado. Éste dio prioridad a las tareas en base en lo implementado y puedo suspenderlas a favor de otra de mayor prioridad, asegurando que procese la tarea más importante primero.

2.2.5. Conclusiones

El control electrónico de servomecanismos mediante microcontroladores es una sencilla y eficaz herramienta que permite un control con alto grado de precisión y velocidad, además con la incorporación de drivers adecuados es posible manejar motores de grandes cargas. Los motores a pasos son capaces de sustituir motores convencionales de CD así como servomotores, ya que con esta mecanismo es posible realizar tanto el control de velocidad como de posicionamiento tal como se hace con cada uno de estos tipos de motores.

3. APÉNDICES

Resumen: *En este capítulo se presenta la recopilación de reportes de los proyectos presentados.*

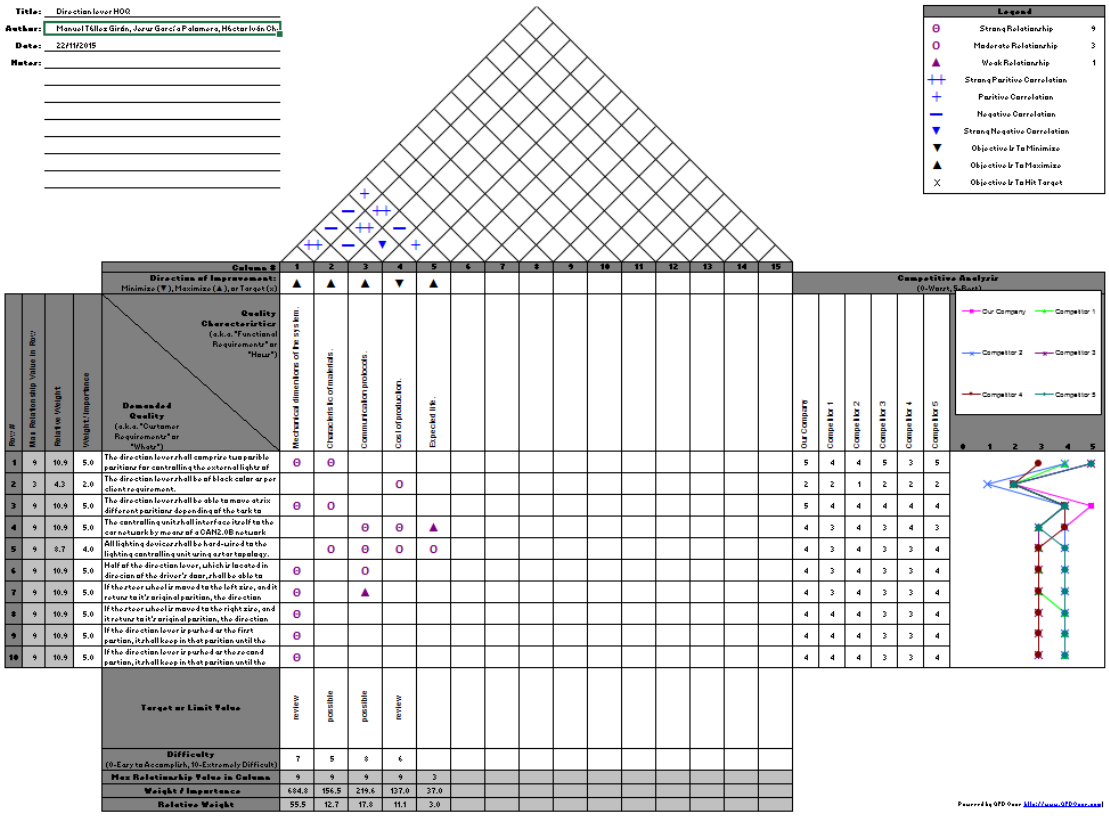
3.1. SISTEMA DE CONTROL DE LUCES REFERENCIAS

Requerimientos de sistema y software

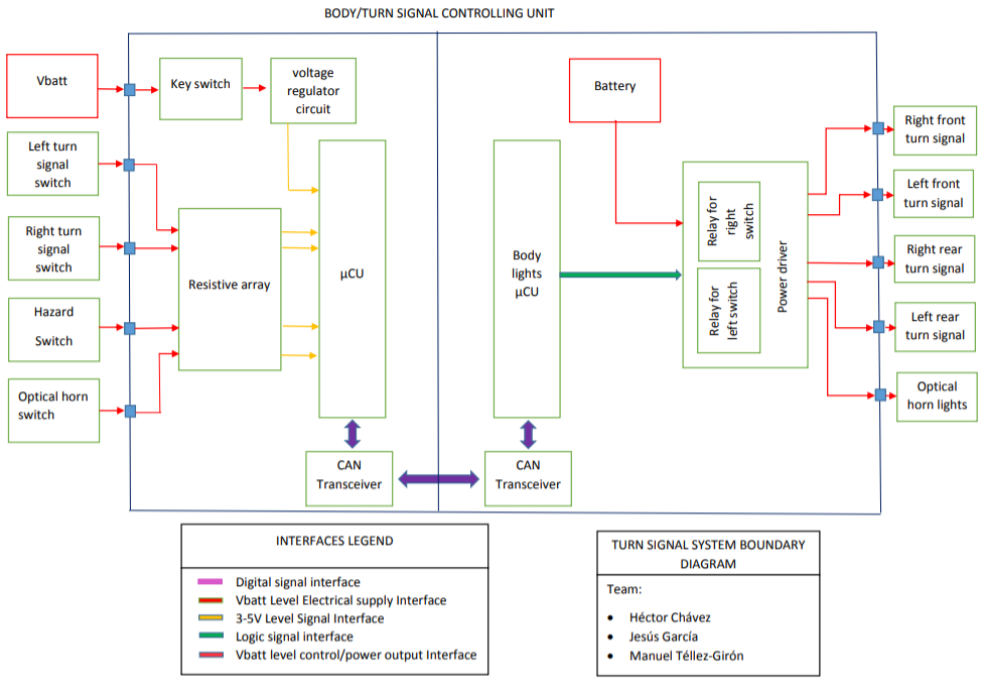
Project Name	Direction lever.											
Students:	*Manuel Tellez Giron. *Jesus Garcia Palamera. *Héctar Iván Chávez Rodríguez.											
		Necesary	Unambiguous	Feasible	Implementation free	Complete	Concise	Consistent	Maintainable	Traceable to origin	Verifiable
DOORS Id	Section	Requirements										
		SYSTEM REQUIREMENTS										
N/A	SYSTEM	The direction lever **shall** comprise four parible paritions for controlling the external lights of the vehicle.	T	T	T	T	T	T	T	T	T	T
N/A	SYSTEM	The direction lever **shall** be of black color as per client requirement.	T	T	T	T	T	T	T	T	T	T
N/A	SYSTEM	The controlling unit **shall** interface itself to the car network by means of a CAN2.0B network working at 500KBps.	T	T	T	T	T	T	T	T	T	T
N/A	SYSTEM	All lighting devices **shall** be hard-wired to the lighting controlling unit using a star topology.	T	T	T	T	T	T	T	T	T	T
N/A	SYSTEM	The direction lever **shall** be able to move at four different positions depending of the task to perform.	T	T	T	T	T	T	T	T	T	T
N/A	SYSTEM	The system **shall** have four switches or sections below:	T	T	T	T	T	T	T	T	T	T
N/A	SYSTEM	1.- One of the four switches, **shall** turn on the left directional headlights of the vehicle.	T	T	T	T	T	T	T	T	T	T
N/A	SYSTEM	2.-One of the four switches **shall** turn on the right directional headlights of the vehicle.	T	T	T	T	T	T	T	T	T	T
N/A	SYSTEM	3.-One of the four switches, **shall** turn on the optical light horn	T	T	T	T	T	T	T	T	T	T
N/A	SYSTEM	4.-One of the four switches, **shall** turn on the hazard lights.	T	T	T	T	T	T	T	T	T	T
N/A	SYSTEM	If the vechicle is turn off and the switch of the left directional is activated, the left directional headlight **shall** not turn on	T	T	T	T	T	T	T	T	T	T
N/A	SYSTEM	If the vechicle is turn on and the switch of the left directional is activated, it **shall** turn on the left directional headlights of the vehicle.	T	T	T	T	T	T	T	T	T	T
N/A	SYSTEM	If the vechicle is turn off and if the switch of the right is activated, the right directional headlight **shall** not turn on	T	T	T	T	T	T	T	T	T	T
N/A	SYSTEM	If the vechicle is turn on and if the switch of the right directional is activated, it **shall** turn on the right directional headlights of the vehicle.	T	T	T	T	T	T	T	T	T	T
N/A	SYSTEM	If the vechicle is turn off and the switch of hazard is activated, the right and left headlights **shall** be turned on	T	T	T	T	T	T	T	T	T	T
N/A	SYSTEM	If the vechicle is turn ON and the switch of hazard is activated, the hazard lights **shall** be turned on	T	T	T	T	T	T	T	T	T	T
N/A	SYSTEM	if the hazard switch is activated and then the system is turned off the hazard lights are kept activated	T	T	T	T	T	T	T	T	T	T
N/A	SYSTEM	If hazard switch is activated, the right and left headlight **shall** be toggling every second	T	T	T	T	T	T	T	T	T	T
			T	T	T	T	T	T	T	T	T	T
		ELECTRICAL REQUIREMENTS										
N/A	ELECTRICAL	The second partition shall turn on the left directional tell-tale of the clurther of the vehicle.	T	T	T	T	T	T	T	T	T	T
N/A	ELECTRICAL	The second partition shall turn on the left directional headlights of the vehicle.	T	T	T	T	T	T	T	T	T	T
N/A	ELECTRICAL	The second partition shall turn on the left directional taillights of the vehicle.	T	T	T	T	T	T	T	T	T	T
N/A	ELECTRICAL	The third partition shall turn on the right directional tell-tale of the clurther of the vehicle.	T	T	T	T	T	T	T	T	T	T
N/A	ELECTRICAL	The third partition shall turn on the right directional headlights of the vehicle.	T	T	T	T	T	T	T	T	T	T
N/A	ELECTRICAL	The third partition shall turn on the right directional taillights of the vehicle.	T	T	T	T	T	T	T	T	T	T
N/A	ELECTRICAL	The fourth partition shall turn on the high beam headlights tell-tale of the clurther of the vehicle.	T	T	T	T	T	T	T	T	T	T
N/A	ELECTRICAL	The fourth partition shall turn on the high beam headlights of the vehicle.	T	T	T	T	T	T	T	T	T	T
N/A	ELECTRICAL	The fifth partition shall turn on the high beam headlights tell-tale of the clurther of the vehicle until the direction lever return to the first position.	T	T	T	T	T	T	T	T	T	T
N/A	ELECTRICAL	The fifth partition shall turn on the high beam headlights of the vehicle until the direction lever return to the first position.	T	T	T	T	T	T	T	T	T	T
N/A	ELECTRICAL	The sixth partition shall turn on the low beam headlight tell-tail of the clurther of the vehicle.	T	T	T	T	T	T	T	T	T	T
N/A	ELECTRICAL	The sixth partition shall turn on the low beam headlights of the vehicle.	T	T	T	T	T	T	T	T	T	T
N/A	ELECTRICAL	If the the direction lever is on the first position (OFF state), it shall do anything and the direccional tell-tails of the clurther of the car shall be turned off	T	T	T	T	T	T	T	T	T	T
N/A	ELECTRICAL	If the the direction lever is on the first position (OFF state), it shall do anything and the direccional headlights of the vehicle shall be turned off	T	T	T	T	T	T	T	T	T	T
N/A	ELECTRICAL	If the the direction lever is on the first position (OFF state), it shall do anything and the direccional taillights of the vehicle shall be turned off	T	T	T	T	T	T	T	T	T	T
N/A	ELECTRICAL	If the the direction lever position is on the second position, the right directional tell-tail of the clurther of the vehicle shall be toggling every second.	T	T	T	T	T	T	T	T	T	T
N/A	ELECTRICAL	If the direction lever position is on the third position, the left directional tell-tail of the clurther of the vehicle shall be toggling every second,	T	T	T	T	T	T	T	T	T	T
			T	T	T	T	T	T	T	T	T	T

	*Manuel Tellez Giron. *Jesus Garcia Palomera. *Héctor Iván Chávez Rodríguez.										
		Necesary	Unambiguous	Feasible Implementation time	Complete	Concise	Consistent	Maintainable	Traceable to origin	Verifiable
Section	Requirements										
	SOFTWARE REQUIREMENTS										
SOFTWARE	The ECU1 **shall** support three digital inputs as are shown bellow:	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	1.- Action left (input).	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	2.-Action right (input).	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	3.- Action hazard lights (input).	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	4- Optical horn switch (input).	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	The ECU2 **shall** support three digital outputs: as are shown bellow:	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	1.- Fault light's Telltail (output).	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	2.- Right headlight (output).	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	3.-Left headlight (output).	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	4.-Optical horn light (output)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	The system communication **shall** support CAN HS with basic data frame.	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	The system communication **shall** be reprogrammed via CAN HS.	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	The message identifiers CAN device **shall** be of 0x1FF.	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	The system communication **shall** support three kind of CAN messages(Remote type messages and data messages) as follows below:	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	1.- CAN message of diagnostic session.	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	2.-Tester present CAN message.	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	3.-Service CAN message.	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	4.-CAN message input/output of the vehicle.	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	The system communication **shall** have the following diagnostic services:	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	Reading remote messages:	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	1.-Reading status request of software version.	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	2.-Reading status request of left headlight.	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	3.-Reading status request of right headlight.	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	4.-Reading status request of fault light's Telltail.	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	5.-Reading status request of hazard lights.	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	Writting remote messages:	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	1.-Writting status request of left headlights.	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	2.-Writting status request of right headlight.	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	3.-Writting status request of fault light's Tell-tail.	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	4.-Writting status request of hazard lights.	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	5.-Reading data messages:	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	Data messages:	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	Reading data messages:	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	1.-Reading status report of left headlights.	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	2.-Reading status report of right headlight.	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	3.-Reading status report of fault light's Telltail.	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	4.-Reading status report of hazard lights.	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	5.-Reading status report of hazard lights.	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	Writting data messages:	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	1.-Writting status report of left headlights.	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	2.-Writting status report of right headlight.	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	3.-Writting status report of fault light's Telltail.	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SOFTWARE	4.-Writting status report of hazard lights.	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

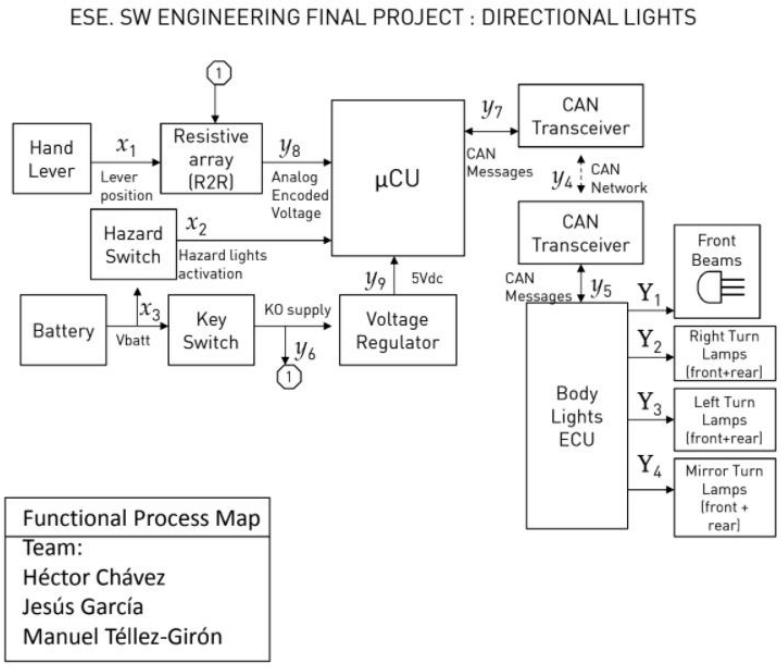
House of quality



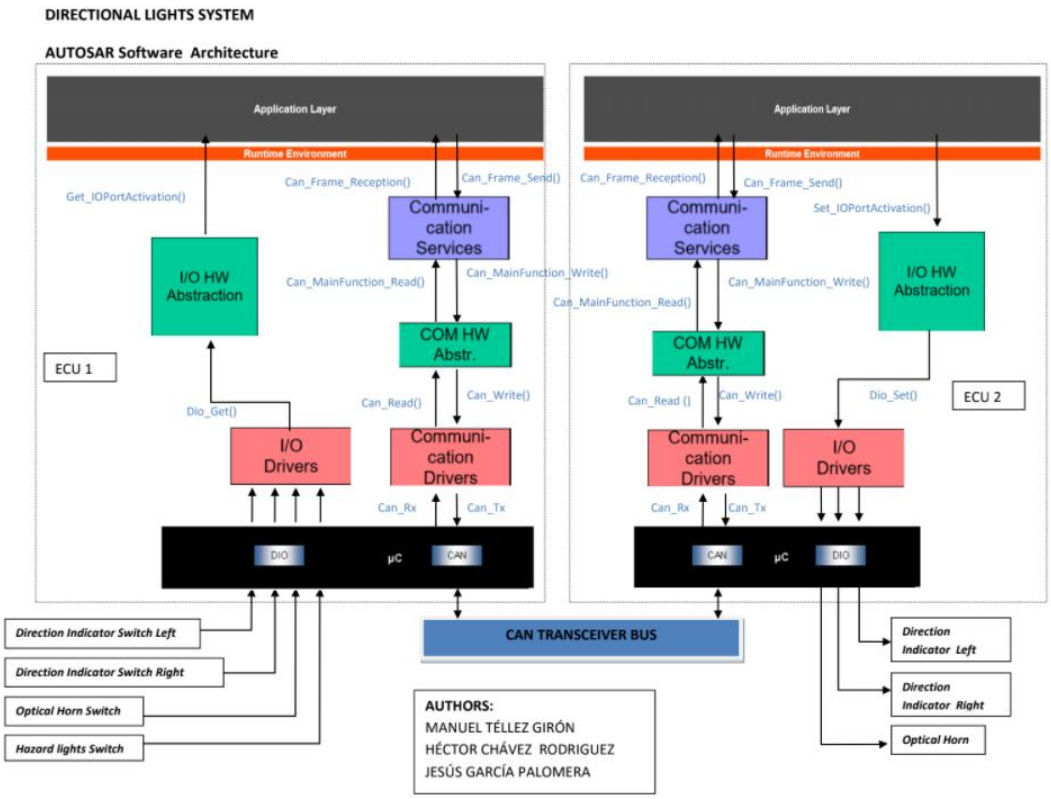
Boundary diagram



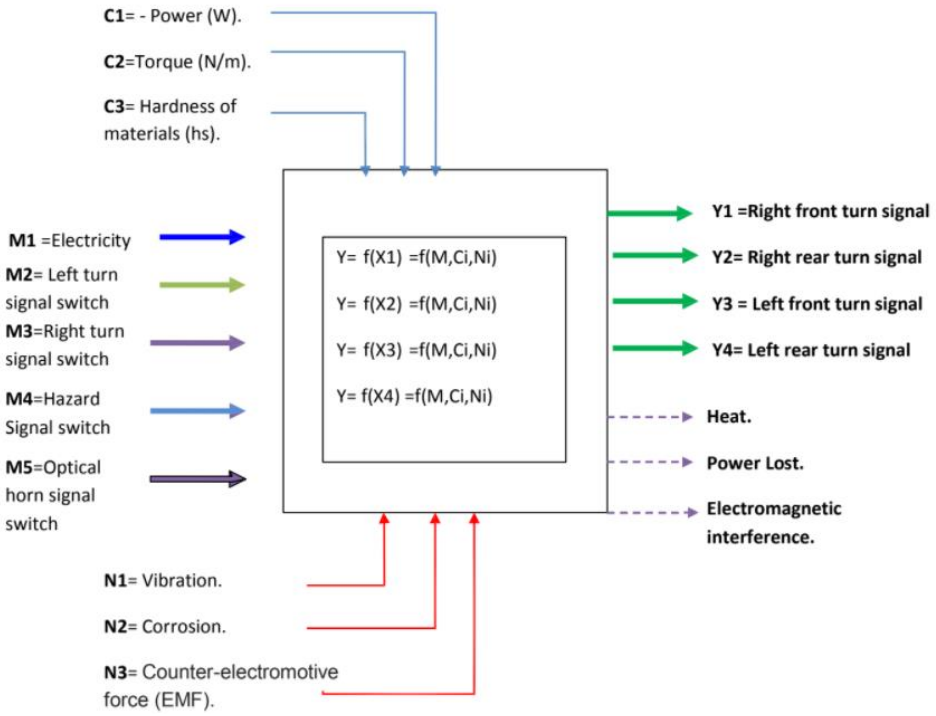
Functional process map



Software AUTOSAR architecture



Parameter Diagram



Software test report

	PROJECT NAME:		BIDIRECTIONAL LIGHTS CONTROL MODULE			
	TEAM:		JESÚS GARCÍA, HÉCTOR CHÁVEZ, MANUEL TÉLLEZ-GIRÓN			
	DATE:		DEC 5TH, 2015			
TEST	TEST CASE NAME	PRE-CONDITIONS OF THE SYSTEM	TEST CASE DESCRIPTION	Expected result	Obtained Result	PASSIFAIL
1	ST1	The vechicle shall be turned on	Enable the right switch in order to turn the right headlight on	The right headlight signal is turned on, and all the missing lights are keep turned off.		
2	ST2	The vechicle shall be turned on	Enable the left switch in order to turn the right headlight on	The left headlight signal is turned on, and all the missing lights are keep turned off.		
3	ST3	The vechicle shall be turned on	Enable the Hazard switch in order to turn both headlights (right and left) on	The left and right headlights are turned on at the same time, and these headlights are toggling every second.		
4	ST4	The vechicle shall be turned off	Enable the right switch while the vehichle is turn off to test the proper functioning of the system.	All the headlights are maintain turned off.		
5	ST5	The vechicle shall be turned off	Enable the left switch while the vehichle is turn off to test the proper functioning of the system.	All the headlights are maintain turned off.		
6	ST6	The vechicle shall be turned off	Enable the Hazard switch while the vechicle is turn off to test the proper functioning of the system.	All the headlights are maintain turned off.		
7	ST7	The vechicle shall be turned on	See the system behavior when the system is in turn on state and the right switch is enabled as well. After a while, the tester shall turn the system off while the right light is kept enabled.	The right headlight is turned on, but soon returns to turn off state due to the system was turned off.		
8	ST8	The vechicle shall be turned on	See the system behavior when the system is in turn on state and the left switch is enabled as well. After a while, the tester shall turn the system off while the left light is kept enabled.	The left headlight is turned on, but soon returns to turn off state due to the system was turned off.		
9	ST9	The vechicle shall be turned on	See the system behavior when the system is in turn on state and the Hazard switch is enabled as well. After a while, the tester shall turn the system off while the hazard lights are kept enabled.	The left and right headlight are turned on and they are toggling every second, both headlights are toggling every second and are kept in that state.		
10	ST10	The vechicle shall be turned off	See the system behavior when the system is in turn off state and the right switch is enabled as well. After a while, the tester shall turn the system on while the right light is kept disabled.	The right headlight is in turn off state, but when the system is turned on, the headlight is turned on as well.		
11	ST11	The vechicle shall be turned off	See the system behavior when the system is in turn off state and the left switch is enabled as well. After a while, the tester shall turn the system on while the left lights are enabled.	The left headlight is in turn off state, but when the system is turned on, the headlight is turned on as well.		
12	ST12	The vechicle shall be turned off	See the system behavior when the system is in turn off state and the Hazard switch is enabled as well. After a while, the tester shall turn the system on while the Hazard lights are enabled.	The left and right headlight are in turn off state, but when the system is turned on, both headlights are toggling every second and are kept in that state.		

FAILURE MODE EFFECTS ANALYSIS

FAILURE MODE AND EFFECTS ANALYSIS

Item: SW ENGINEERING FINAL PROJECT Responsibility:
Model: DIRECTIONAL LIGHTS SYSTEM Prepared by:
Core Team: JESUS GARCIA, HECTOR CHÁVEZ, MANUEL TELLEZ-GIRÓN

Process Function (What is the process step/output?)	Potential Failure Mode (What can go wrong with the process step/output?	Potential Effect(s) of Failure (What is the impact on the customer (output variables) or internal requirements?)	Sev	Potential Cause(s)/ Mechanism(s) of Failure (What are the root cause reasons for the process step/input to go wrong? These are the x's)	Occur	Current Process Controls (What are the existing controls that prevent or detect either the cause or the FM prior to leaving the process step?)	Detec	RPN	Recommended Action(s) (What are the actions for reducing the OCC. Of the cause or improving DET?)
Hand lever	Hand lever will not move/ stuck	Actuation signal will not be generated	1	broken mechanical pieces due to life-cycles end/ wear	1	DFR process, materials theoretical modeling	3	3	Perform sufficient life testing and mechanical/electrical cycles testing , materials degradation and bathtub (gamma functions) modeling
Hand lever	Hand lever will not send the right electrical signal	Actuation signal will not be generated	1	Contacts worn	3	Right seleccion of metal alloys at design time	3	9	Materials environmentals tests: humidity, temperature, salinity at rated bathtub life-cycles
Hand lever	Hand lever will not send the right electrical signal	Actuation signal will not be generated	1	Contacts dirty or contaminated with debris	1	enclosure of hermeticity and dust-proof considerations	5	5	Hermeticity testing, dust blast testing
Resistive array	Electrical analog encoded signal is out of range	Actuation signal will be misinterpreted and lights will actuate erroneusly	3	Resistive values derating due to environmental long term effects	3	Resistor design binning and WCA	7	63	Life testing and different WCAs using Montecarlo and statistical modelings
Battery	Electrical power is insufficient and causes erratical behavior	Lighting system will behave randomly and several and unpredictable failures appear	9	End of life of battery, potential failures in electrical system	3	Design is fault tolerant and Brownout halts are considered	1	27	DOEs and extended endurance testing at electrical system levels.
Hazard switch	Hazard signal does not turn at button actuation	At an emergency event hazard is not notified to other drivers	9	Mechanical failures of button	1	DFR process, materials theoretical modeling	1	9	Perform sufficient life testing and mechanical/electrical cycles testing , materials degradation and bathtub (gamma functions) modeling
Hazard switch	Hazard signal does not turn at button actuation	At an emergency event hazard is not notified to other drivers	9	Electrical wear of contacts, false contacts	3	DFR process, materials theoretical modeling	1	27	Perform sufficient life testing and mechanical/electrical cycles testing , materials degradation and bathtub (gamma functions) modeling
Hazard switch	Hazard signal does not turn at button actuation	At an emergency event hazard is not notified to other drivers	9	Unintended disconnection of fast-on connectors	3	Stress design and testing	5	135	Harnesses cross section testing at Realiability testing, pull testing, vibration testing
Key switch	Directional light system is not properly powered	Power is not properly supplied to light system (intermittent, poor contact , false or no contact)	9	Unintended disconnection of fast-on connectors	1	Stress design and testing	5	45	Harnesses cross section testing at Realiability testing, pull testing, vibration testing
Voltage Regulator	Supplied voltage is very above specifications	Electronic circuitry permanent damage and overall system malfunction	9	Thermal derating of regulator or consistent overstressing of device	1	Standard WCA at maximum DC level of Vbatt.	7	63	Transient modeling and testing, EMI testing, EMC testing, dip and stress electrical testing
Voltage Regulator	Supplied voltage is below operational level	Electronic circuitry malfunction and unpexpected behaviors	9	Thermal derating of regulator or consistent overstressing of device	1	Standard WCA at minimum DC level of Vbatt.	7	63	Transient modeling and testing, EMI testing, EMC testing, dip and stress electrical testing
MCU	System behavior is wrong, erratic, or does not obbey to actuation commands	MCU presents abnormal functions	9	Thermal (overheat) or electrical (dips, transients, FLASH memory unintended erasure, clock halting) root causes lead to MCU permanent or temporary malfunction	3	Code module testing, system testing, WCA, DV/PV, Reliability tests	5	135	Transient modeling and testing, EMI testing, EMC testing, dip and stress electrical testing, Radiation testing and FLASH memory endurance testing at system environment
CAN Transceivers	Wrong messages, corrupted messages, not matching ids/CRCs	Lights will not obbey to commands, wrong lights actuation	7	Wrong selection or configuration of CAN transceivers	1	CAN Breadboarding, bus characterization, WCA, bus modeling, EMI Tests	3	21	DOE /FRD design of experiments in presence of noise, system modeling at several environmental conditions
CAN Transceivers	Wrong messages, corrupted messages, not matching ids/CRCs	Lights will not obbey to commands, wrong lights actuation	7	Contact failures at wiring/harnesses levels	3	Sectional testing, pull/insertion tests	3	63	Endurance and life testing of wires
Lights body controller	Some lights will not turn off/on as expected	Malfunction of body controller	7	Thermal or environmental failures at system level	1	Module test, system test scoping body controller and directional lights module separatedly	5	35	System level testing using DOE/FRD tests. Boss modeling tests in presence of noise
Lights body controller	Some lights will not turn off/on as expected	False,poor or no appropiate contact of fast-on connectors	7	Contact failures at wiring/harnesses levels	2	Sectional testing, pull/insertion tests	5	70	Endurance and life testing of wires, thermal tests and after-cycling thermal sectional tests

3.2. CONTROL DE LA VELOCIDAD DE UN MOTOR BIPOLAR A PASOS REFERENCIAS

ITESO, Universidad Jesuita de Guadalajara

Departamento de Electrónica, Sistemas e Informática

Especialidad en Sistemas Embebidos

EMBEDDED SYSTEMS POSTGRADUATE PROGRAM

DESI, ITESO A.C.

Control de la velocidad de un motor bipolar a pasos.

Integrantes:

Héctor Iván Chávez Rodríguez
Jesus Garcia Palomera

RESUMEN:

El trabajo consistió en la realización de un controlador digital para la regulación de la velocidad de un motor bipolar a pasos por medio de la programación en lenguaje C de un Microcontrolador PIC de Microchip usando PICOS18 como sistema operativo en tiempo real.

Para el logro del objetivo propuesto, se estudió el principio de funcionamiento de diversos driver controladores de motores así como el de un motor a pasos, posteriormente se hizo uso de las APIS que nos ofrece el kernel de PICOS18 para poder realizar la practica mencionada.

INTRODUCCIÓN:

En la actualidad existen numerosas aplicaciones tanto industriales como académicas en las que se exige un control preciso y exacto de los actuadores implementados en tareas de transmisión de movimiento, posicionamiento y control de velocidad.

Existen numerosas tarjetas de control para este tipo de motores, sin embargo el costo y programación de las mismas en ocasiones complican la aplicación dentro de los proyectos donde se incorporan microcontroladores. La aplicación de un controlador que permita la variación de las señales al motor con altas frecuencias y a la vez realizar diferentes tareas como es la comunicación y control de otros elementos es de gran importancia.

FUNDAMENTOS TEORICOS:

Motor a pasos

Un motor a pasos, como todo motor, es en esencia un conversor electromecánico, que transforma energía eléctrica en mecánica. Mientras que un motor convencional gira libremente al aplicarle una tensión, el motor paso a paso gira un determinado ángulo de forma incremental (transforma impulsos eléctricos en movimientos de giro controlados), lo que le permite realizar desplazamientos angulares fijos muy precisos (pueden variar desde 1,80° hasta unos 90°).

Este tipo de motores son ideales cuando lo que queremos es posicionamiento con un elevado grado de exactitud y/o una muy buena regulación de la velocidad.

Están constituidos esencialmente por dos partes:

-**Estator:** parte fija construida a base de cavidades en las que van depositadas las bobinas.

-**Rotor:** parte móvil construida mediante un imán permanente.

Figura 1.- Diagrama interno motor a pasos.

Principio de funcionamiento:

Al excitar el estator, se crearan los polos N-S, provocando la variación del campo magnético formado. La respuesta del rotor será seguir el movimiento de dicho campo (tenderá a buscar la posición de equilibrio magnético), es decir, orientará sus polos NORTE-SUR hacia los polos SUR-NORTE del estator, respectivamente. Cuando el rotor alcanza esta posición de equilibrio, el estator cambia la orientación de sus polos y se tratará de buscar la nueva posición de equilibrio. Manteniendo dicha situación de manera continuada, se conseguirá un movimiento giratorio y continuo del rotor, produciéndose de este modo el giro del eje del motor, y a la vez la transformación de una energía eléctrica en otra mecánica en forma de movimiento circular.

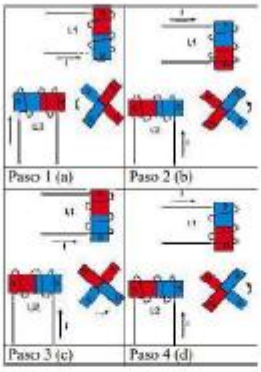


Figura 2.-Secuencia de polarización de un motor a pasos.

Motores Bipolares: En este tipo de motores las bobinas del estator se conectan en serie formando solamente dos grupos, que se montan sobre dos estatores, tal y como se muestra:

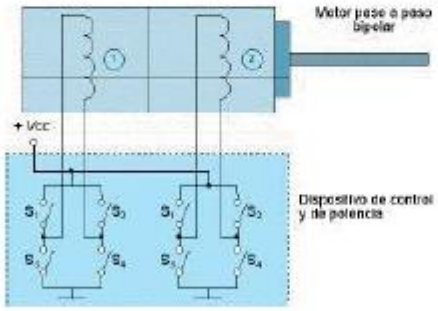


Figura 3.-Diagrama de control motor paso a paso bipolar.

MATERIALES Y MÉTODOS.

El proyecto consintió en la realización del control de la velocidad de un motor bipolar a pasos, en el cual por medio de por dos botones se hizo el incremento o decremento de la velocidad. Para esto se hizo uso de las interrupciones externas RB1 y RB2, para lo cual ambas se configuraron como interrupciones externas en el archivo main.c dentro de la función init como se muestra a continuación:

```
void Init(void)
{
    FSR0H = 0;
    FSR0L = 0;

    /* User setting : actual DIC frequency */
    Tmr0.lt = TMR0_PRESET;

    /* Timer OFF - Enabled by Kernel */
    T0CON = 0x08;
    TMR0H = Tmr0.bt[1];
    TMR0L = Tmr0.bt[0];

     OSCCON = 0b01110000;

    PIR1 = 0;
    PIR2 = 0;
    RCON = 0;
    IPR1 = 0;
    IPR2 = 0;

    INTCON3bits.INT1IP = 0;
    INTCON3bits.INT2IP = 0;
    INTCON3bits.INT1IE = 1;
    INTCON3bits.INT2IE = 1;
    TRISBbits.TRISB1 = 1;
    TRISBbits.TRISB2 = 1;
    TRISDbits.TRISD3 = 1;

    TRISDbits.TRISD0 = 0;
    TRISDbits.TRISD7 = 0;
    //configure buttons
    WPUB = 0b00011111;
    INTCON2bits.RBPU = 0;                    //
}
```

Figura 4.-Configuración de las resistencias pull ups

e interrupciones de baja prioridad.

Posteriormente se crearon por medio de la API SetRelAlarm los estados necesarios para crear 4 PWM que serán aplicados a cada terminal que se aplicara al motor como se muestra en la siguiente figura.

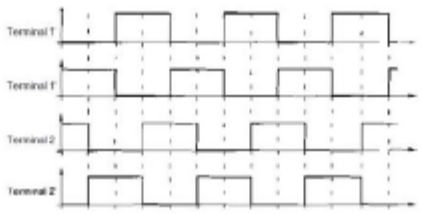


Figura 5.-Secuencia de señal de excitación del motor paso a paso bipolar.

Paso	L1	L1'	L2	L2'
1	-	+	+	-
2	-	+	-	+
3	+	-	-	+
4	+	-	+	-

Figura 6.-Secuencia de excitación de las bobinas del motor paso a paso bipolar.

El código utilizado para generar las señales presentadas se muestra a continuación:

```
TASK(TASK0)
{
    TRISDbits.TRISD2 = 0;
    TRISDbits.TRISD3 = 0;
    TRISDbits.TRISD4 = 0;
    TRISDbits.TRISD5 = 0;

    while(1)
    {
        SetRelAlarm(ALARM_TSK0, 1000, 30);
        WaitEvent(ALARM_EVENT);
        ClearEvent(ALARM_EVENT);

        counter++;

        if ((counter == duty) || (counter == period))
        {
            if(step == 1)
            {
                LATDbits.LATD2 = 1;
                LATDbits.LATD3 = 0;
                LATDbits.LATD4 = 0;
                LATDbits.LATD5 = 1;
                step ++;
            }
            else if(step == 2)
            {
                LATDbits.LATD2 = 0;
                LATDbits.LATD3 = 1;
                LATDbits.LATD4 = 0;
                LATDbits.LATD5 = 1;
                step ++;
            }
            else if (step == 3)
            {
                LATDbits.LATD2 = 0;
                LATDbits.LATD3 = 1;
                LATDbits.LATD4 = 1;
                LATDbits.LATD5 = 0;
                step ++;
            }
            else if (step == 4)
            {
                LATDbits.LATD2 = 1;
                LATDbits.LATD3 = 0;
                LATDbits.LATD4 = 1;
                LATDbits.LATD5 = 0;
                step ++;
            }
            else if (step == 5)
            {
                step = 1;
            }
            if (counter == period)
            {
                counter = 0;
            }
        }
    }
}
```

Figura 7.-Codigo implementado para la secuencia de conmutación.

A continuación se muestra el código encargado del control de la velocidad del motor:

```
void MyOwnISR(void)
{
    //configure buttons
    NPUB = 0b00011111;
    INTCON2bits.RBPU = 0;

    if (INTCON3bits.INT1IF)
    {
          LATDbits.LATD0 = ~LATDbits.LATD0;
            if (duty >= 0)
            {
                duty = duty - 2;
            }

    INTCON3bits.INT1IF = 0;
    }

    if (INTCON3bits.INT2IF)
    {
          LATDbits.LATD7 = ~LATDbits.LATD7;
          if (duty <= period)
          {
              duty = duty + 2;
          }

       INTCON3bits.INT2IF = 0;
    }
}
```

Figura 8.-Codigo implementado dentro de la función de interrupción.

Al ser presionados los botones encargados de generar la interrupción, validaran el estado de la variable duty y period. Al presionar el botón RB2, el valor de duty será incrementado y esta aumentara el duty cycle por lo que aumentara su velocidad, y lo mismo procedimiento pasara al ser presionado el botón RB1 con la diferencia de que esta decrementara el valor de duty y por consiguiente su velocidad decrementara.

Posteriormente se realizó la conexión entre el Microcontrolador a un puente H de transistores el cual permite amplificar la corriente que se le aplicara al motor, el diagrama se muestra a continuación.

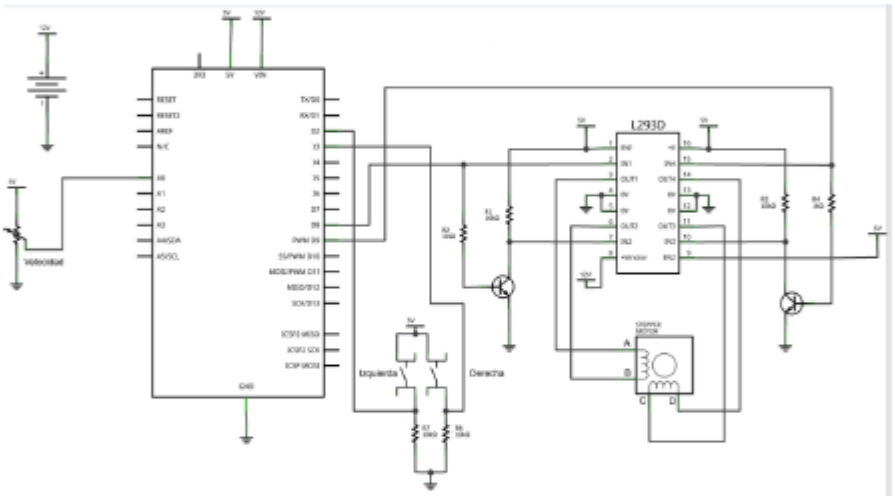


Figura 9.-Diagrama de control motor paso a paso.

CONCLUSIONES.

El control electrónico de servomecanismos mediante microcontroladores es una sencilla y eficaz herramienta que permite un control con alto grado de precisión y velocidad, además con la incorporación de drivers adecuados es posible manejar motores de grandes cargas.

Los motores a pasos son capaces de sustituir motores convencionales de CD así como servomotores, ya que con esta mecanismo es posible realizar tanto el control de velocidad como de posicionamiento tal como se hace con cada uno de estos tipos de motores.

TABLA DE DOCUMENTACIÓN

REQUERIMIENTOS DE SISTEMA Y SW	Microsoft Excel 97-2003 Worksheet
HOUSE OF QUALITY	Microsoft Excel 97-2003 Worksheet
BOUNDARY DIAGRAM	Adobe Acrobat Document
FUNCTIONAL PROCESS DIAGRAM	Adobe Acrobat Document
SOFTWARE AUTOSAR ARCHITECTURE	Adobe Acrobat Document
PARAMETER DIAGRAM	Adobe Acrobat Document
SOFTWARE TEST REPORT	Microsoft Excel 97-2003 Worksheet
FMEA	Adobe Acrobat Document
TABLA DE LA VERDAD	Microsoft Excel 97-2003 Worksheet
PROYECTO CONTROL MOTOR BIPOLAR	Adobe Acrobat Document