



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TÍTULO:

**DISEÑO DE UN SISTEMA DE PUNTUACIÓN ELECTRÓNICO
UTILIZANDO COMUNICACIÓN INALÁMBRICA ZIGBEE E
IMPLEMENTACIÓN DE UN CONTADOR DESCENDENTE PARA LAS
CATEGORÍA ROBOT DE BATALLA Y SUMO.**

AUTOR:

GINE WADIH CONTRERAS FARAH

Previa la obtención del Título

INGENIERO EN TELECOMUNICACIONES

TUTOR:

M. Sc. Daniel Bayardo Bohórquez Escobar

Guayaquil, Ecuador

2016



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

CERTIFICACIÓN

Certificamos que el presente trabajo fue realizado en su totalidad por el Sr.
Gine Wadih Contreras Farah como requerimiento parcial para la obtención
del título de INGENIERO EN TELECOMUNICACIONES.

TUTOR

M. Sc. Daniel Bayardo Bohórquez Escobar

DIRECTOR DE CARRERA

M. Sc. Miguel A. Heras Sánchez.

Guayaquil, a los 14 del mes de Marzo del año 2016



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

Yo, **Gine Wadih Contreras Farah**

DECLARÓ QUE:

El trabajo de titulación “**Diseño de un Sistema de Puntuación Electrónico utilizando Comunicación Inalámbrica Zigbee e Implementación de un Contador Descendente para las Categorías Robot de Batalla y Sumo**” previa a la obtención del Título de Ingeniero en Telecomunicaciones, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía. Consecuentemente este trabajo es de nuestra autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del Trabajo de Titulación referido.

Guayaquil, a los 14 del mes de Marzo del año 2016

EL AUTOR

GINE WADIH CONTRERAS FARAH



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

AUTORIZACIÓN

Yo, **Gine Wadih Contreras Farah**

Autorizamos a la Universidad Católica de Santiago de Guayaquil, la publicación, en la biblioteca de la institución del Trabajo de Titulación: “**Diseño de un Sistema de Puntuación Electrónico utilizando Comunicación Inalámbrica Zigbee e Implementación de un Contador Descendente para las Categorías Robot de Batalla y Sumo**”, cuyo contenido, ideas y criterios es de mi exclusiva responsabilidad y autoría.

Guayaquil, a los 14 del mes de Marzo del año 2016

EL AUTOR

GINE WADIH CONTRERAS FARAH

DEDICATORIA

Esta tesis y más que nada esta carrera quiero dedicársela a Dios, a mi madre, a mis hermanos, a mi novia que han sido un pilar tan fundamental en este proceso pero especialmente quiero dedicárselos a mis abuelos que son los que día a día me dan esas ganas de seguir adelante.

EL AUTOR

GINE WADIIH CONTRERAS FARAH

AGRADECIMIENTO

Quiero agradecer especialmente a mis abuelos por cada uno de los sacrificios que han hecho para que esto sea posible, a Dios por brindarme la perseverancia día a día, a mi madre por enseñarme que se puede conseguir las cosas que uno se propone y a mi novia que es la que día a día me da ese empuje para seguir avanzando.

EL AUTOR

GINE WADIH CONTRERAS FARAH

Índice General

Índice de Figuras	X
Índice de Tablas	XII
Resumen	XIII
CAPÍTULO 1: INTRODUCCIÓN	14
1.1. Introducción referente a los Robots de Batalla	14
1.2. Antecedentes	17
1.3. Justificación del Problema	17
1.4. Definición del Problema	18
1.4.1. Objetivo General	18
1.4.2. Objetivos Específicos	18
1.5. Objetivos del Problema de Investigación	19
1.6. Hipótesis	19
1.7. Metodología de Investigación	19
CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA	21
2.1. Introducción a la microelectrónica	21
2.1.1. Influencia económica de los semiconductores I	22
2.1.2. Impacto de los semiconductores en los bienes de consumo	22
2.1.3. Circuito integrado de aplicación específica (ASIC)	24
2.1.4. Lógica de campo programable	24
2.1.6. Componentes virtuales	25
2.1.7. Sistema de automatización de diseño electrónico de nivel ...	26
2.1.8. Diseño a nivel de sistema	27
2.1.9. El diseño de algoritmos	27

2.2.	Historia de los microcontroladores	28
2.2.1.	1947: Punto de Contacto de transistor de germanio	28
2.2.2.	Invencción del transistor	29
2.2.3.	Invencción Europea del transistor.....	29
2.2.4.	Transistron.....	30
2.2.5.	Comercialización del Transistor	30
2.2.6.	Silicon Valley- Fairchild.....	31
2.2.7.	Transistor MOS.....	32
2.3.	Microcontroladores.....	33
2.3.1.	Términos usados frecuentemente.....	34
2.3.2.	¿Cuál es la diferencia entre un microprocesador y un microcontrolador?	37
2.3.3.	Familia de los microcontroladores	37
2.3.4.	Fabricantes de microcontroladores	38
2.4.	Microcontrolador de 8 bits	38
2.4.1.	Diagrama de bloques de microprocesador de 8 bit.....	39
2.4.2.	¿Qué es un microcontrolador de 8 bits?	40
2.4.3.	Tipos de microcontroladores de 8 bits	40
2.4.4.	Aplicaciones para los microcontroladores de 8 bits:.....	41
2.5.	Microcontrolador de 16 bits	42
2.5.1.	Diagrama de bloques de un microcontrolador de 16 bits	42
2.5.2.	¿Qué es un microcontrolador de 16 bits?	43
2.5.3.	Tipos de Microcontroladores de 16 bits	43
2.5.4.	Aplicaciones para microcontroladores de 16 bits	44
2.5.5.	Microcontrolador de 32 bits.....	45

2.5.6.	Diagrama de bloque de un microcontrolador de 32 bits.	45
2.5.7.	¿Qué es un microcontrolador de 32 bits?	46
2.5.8.	Tipos de microcontroladores de 32 bits	46
2.5.9.	Aplicaciones para microcontroladores de 32 bits	47
2.6.	Microcontrolador PIC 18F4550.....	47
2.6.1.	Diagramas de Pines	47
 CAPÍTULO 3: SIMULACION Y RESULTADOS OBTENIDOS.....		50
3.1.	Diseño del Sistema de Puntuación Electrónico.	50
3.2.1.	Diseño del Sistema de Puntuación Electrónico.....	50
3.2.2.	Programación de alto nivel en CCS C Compiler para el sistema de marcación electrónica.	53
3.2.3.	Resultados obtenidos del sistema de marcación electrónica en Proteus.	61
3.2.4.	Diseño electrónico PCB del sistema de marcación electrónica en Proteus.	63
3.3.	Diseño del contador descendente.	67
3.3.1.	Diseño del Sistema de Puntuación Electrónico.....	67
 CAPÍTULO 4: CONCLUSIONES Y RECOMENDACIONES.....		71
4.1.	Conclusiones.....	73
4.2.	Recomendaciones.....	74
 REFERENCIAS BIBLIOGRÁFICAS.....		75

Índice de Figuras

Capítulo 2

Figura 2. 1: Dispositivos electrónicos que utilizan semiconductores.	23
Figura 2. 2: Transistor que revolucionó las computadoras.	28
Figura 2. 3: Transistron.....	30
Figura 2. 4: Científicos de Fairchild.....	32
Figura 2. 5: Transistor MOS.....	32
Figura 2. 6: Micro de 8 bits	39
Figura 2. 7: Diagrama de bloques de un micro de 8 bits	39
Figura 2. 8: Micro de 16/32 bit	42
Figura 2. 9: Diagrama de bloques de un micro de 16 bits	42
Figura 2. 10: Micro de 16/32 bit	45
Figura 2. 11: Micro de 16/32 bit	45
Figura 2. 12: Diagrama esquemático del PIC 18F4550 de 40 pines.	48
Figura 2. 13: Diagrama esquemático del PIC 18F4550 de 44 pines.	48
Figura 2. 14: Diagrama de PIC 18f4550.....	49

Capítulo 3

Figura 3. 1: Circuito para el equipo de marcación de puntos para la categoría de robot de batalla.....	51
Figura 3. 2: Descripción del encabezado del programa.	54
Figura 3. 3: Declaración de variables globales.....	54
Figura 3. 4: Función void RDA_isr para identificar el dispositivo y envió de datos acumulados en los registros internos.	55
Figura 3. 5: Función incrementa_cinco, encargada de aumentar el valor de la variable unidad y decena según las circunstancias de la competencia.	56
Figura 3. 6: Función incrementa_diez para el incremento para diez puntos.	56
Figura 3. 7: Función incrementa_veinte para el incremento para veinte puntos.	57

Figura 3. 8: Función decremента, encargada de descontar 5 puntos a los valores que posee actualmente la variable "puntaje".	57
Figura 3. 9: Función principal del código para el sistema de puntuación.	59
Figura 3. 10: Código para efecto de simulación y demostración del sistema de puntuación.	60
Figura 3. 11: Circuito para verificar la simulación del equipo de marcación de puntos para la categoría de robot de batalla.	61
Figura 3. 12: Resultado obtenido para inmovilizar al oponente por ataque. .	62
Figura 3. 13: Resultado obtenido para embestidas.	62
Figura 3. 14: Resultado obtenido para vuelcos.	63
Figura 3. 15: Vista superior del PCB para el sistema de puntuación.	64
Figura 3. 16: Vista inferior del PCB para el sistema de puntuación.	65
Figura 3. 17: Vista de la placa PCB del sistema de puntuación.	66
Figura 3. 18: Circuito contador de tiempo para competencia del sumo	68

Índice de Tablas

Capítulo 2

Tabla 2. 1: Ingresos por vendedores de microcontroladores.....	21
Tabla 2. 2: Listado de semiconductores.....	23

Resumen

El presente trabajo se enfoca en el diseño e implementación de un marcador electrónico inalámbrico con comunicación ZIGBEE para utilizar como sistema de marcación de los jueces en la categoría Robot de Batalla, y de un contador para controlar el tiempo de accionamiento de los robots Sumo. El uso en categoría de batalla, consiste en que el dispositivo es accionado por el jurado, mediante seis pulsadores, que son procesados a través del microcontrolador PIC 18F4550 y finalmente transmitido mediante comunicación inalámbricos ZigBee hasta la computadora, donde se visualizará el resultado obtenido durante la batalla. La ventaja de utilizar ZigBee, es que no requiere de licencia para operar en la banda libre ISM de frecuencia 2.4 GHz. Aunque ZigBee puede operar en 3 bandas de frecuencia. El propósito del trabajo de titulación, fue mejorar la forma en la que el jurado llevaba las puntuaciones de manera manual y esto dificultaba el conteo de puntos en cada robot. Mientras, que para la categoría Sumo, el control del tiempo se utilizaban celulares para ver el tiempo de 5 s en las que los robots debían atacar una vez que fue accionado el pulsador. El lenguaje de programación de alto nivel, que se utilizó fue SCC C Compiler, el software de simulación de microcontroladores es, Isis de Proteus.

CAPÍTULO 1: INTRODUCCIÓN

1.1. Introducción referente a los Robots de Batalla.

Ha habido muchos artículos y libros escritos durante los años tratando de desenredar la oscura prehistoria de combate robótico en el mundo, pero esta vez me referiré a la prehistoria de los combates robóticos en los Estados Unidos. Muchas personas trazan las raíces en las competiciones de ingeniería en el Instituto Tecnológico de Massachusetts (MIT), Instituto de Tecnología de California (CalTech) y otras universidades en la década de 1970, mientras que otros ven a los artistas de performance de San Francisco de la década de 1980 y sus creaciones mecánicas monstruosas.

Para la mayoría de los aficionados, su primer contacto con este deporte fue en 1994 cuando Marc Thorpe organizó las primeras competiciones Robot Wars en San Francisco, California. Pero, Thorpe estaba lejos de ser el primero.

De hecho, fue al menos ocho años antes en la ciudad de Denver que las semillas de combate robótico se posicionaron primero. Un ingeniero con el nombre de Bill Llewellyn, se reunió con un grupo de amigos con ideas afines a la robótica y formó el "Denver Mad Scientists Club" y organizó un concurso donde las personas construyen robots autónomos para completar una carrera sencilla. Él lo llamó el "Critter Crawl", y fue recibido con gran interés.

Como recuerda Bill, "Todo el mundo se acercó a nosotros después y dijeron que el evento fue bastante increíble, pero sería mucho mejor si los robots realmente lucharan entre sí." Y así fue, en 1987, Bill y sus cohortes científicos locos, desarrollaron un simple conjunto de reglas para robots controlados por cable y controladores por radio (RC) y lanzaron el primer "Critter Crunch" en la convención de ciencia ficción "MileHiCon".

Desde entonces este acto se ha celebrado todos los años en la convención, por lo que vendría a ser el evento de combate robótico más antiguo y de mayor duración en el mundo. El sistema de puntuación, se otorgan en 2 categorías: (a) agresión - 5 puntos, y (b) daños - 6 puntos.

Cada juez determinaba el número de puntos para adjudicar a cada combatiente en cada categoría, de acuerdo con las orientaciones generales del jurado. La puntuación máxima posible que recibe un combatiente es 11. Por lo tanto, un solo juez otorgará un total de 11 puntos, y un panel de 3 jueces otorgará un total de 33 puntos.

La puntuación de agresión se basaba en la cantidad relativa de tiempo que cada robot gasta en atacar al otro. Los ataques no tienen que tener éxito para contar los puntos de la agresión, pero la distinción se hará entre persiguiendo a un oponente que huía y estrellarse al azar alrededor de la arena.

Los puntos no serán otorgados para la agresión si un robot es completamente incontrolable o no pueden hacer más que gire en su lugar, incluso si se trata de un ataque. También, permanecer sentado y esperando al oponente para conducir en su arma no cuenta para los puntos de la agresión, incluso si es un arma increíblemente destructiva. El robot debía mostrar movimiento de traslación a su oponente para que pueda ser considerado como una agresión.

A continuación, se muestran los puntos de adjudicación por agresión:

- 5-0: una puntuación 5-0 se concederá únicamente cuando uno de los robots no intenta atacar al otro, y el otro ataca constantemente.
- 4-1: una puntuación de 4-1 se otorgará en el caso de predominio significativo de los ataques de un robot, cuando el otro sólo intentar atacar un par de veces durante el partido.
- 3-2: una puntuación 3-2 se adjudicará cuando:
 - Ambos robots atacan constantemente el otro.
 - Ambos robots sólo atacan al otro durante parte del partido.
 - Ambos robots pasan la mayor parte del partido evitando el uno al otro. En este caso, será a discreción del juez decidir qué robot hizo más intentos de ataque hacia el otro.
 - Un combatiente que ataca a una ruleta de todo el cuerpo (por ejemplo, impulsa intencionalmente dentro del perímetro del arma spinning) se considera automáticamente al agresor y recibe una puntuación de 3.2 en el caso de que cualquiera de los robots

atacan constantemente, o ambos robots se evitan constantemente entre sí.

Finalmente, no podía haber empates en la agresión, para lo cual los jueces debían decidir que robot era más agresivo que el otro.

1.2. Antecedentes.

En todos los Concursos de Robótica Nacionales e incluido el CER (Concurso Ecuatoriano de Robótica) existen varias categorías, tales como, Sumo (Mega autónomo, Mega RC, Minisumo y Microsumo), seguidores de línea (velocista y destreza), robot de Batalla (30 y 120 libras), entre otras.

Para la categoría de Sumo, los robots son activados de acuerdo a la disposición del juez y esperan 5 segundos para que el robot inicie el ataque hacia el robot oponente. Mientras, que para la categoría de Robot de Batalla hay dos jueces que llevan las puntuaciones de los dos robots, considerando los ataques con arma, embestidas, vuelcos, inmovilización y sanciones por comportamiento inadecuado de los participantes.

1.3. Justificación del Problema.

El diseño e implementación de un sistema de puntuación y contador electrónico utilizando comunicación inalámbrica Zigbee para la categoría de robot de batalla funcionara acorde cada equipo vaya obteniendo puntos y así poder enviar la información del dispositivo a un PC y poder visualizar por

pantallas, ya que en competencias pasadas las marcaciones de los puntaje se han venido dando manualmente.

1.4. Definición del Problema.

Muy a menudo en los eventos o competencias de robótica que se han venido dando en Ecuador, hemos palpado la necesidad de diseñar e implementar un sistema de puntuación electrónico utilizando comunicación inalámbrica Zigbee y de un contador descendente que hace la función de un cronometro de 5 segundo, el primero debido a que muchas veces los marcadores o las marcaciones de los puntajes en estas competencias han sido de manera manual, sin necesidad de ningún dispositivo electrónico; y el segundo porque el conteo de los 5 segundos ha sido a través de un reloj personal o conteo verbal.

1.4.1. Objetivo General.

Diseñar e Implementar un Sistema de Puntuación Electrónico utilizando Comunicación Inalámbrica Zigbee y de un Contador Descendente para las Categorías Robot de Batalla y Sumo.

1.4.2. Objetivos Específicos.

- Caracterizar los fundamentos teóricos que son la parte fundamental que dan soporte al trabajo de titulación.
- Diseñar a través de Proteus los sistemas de puntuación electrónica y del contador descendente para las categorías de batalla y sumo.

- Realizar los algoritmos de programación en Micro C para los sistemas de marcación y contador descendente.

1.5. Objetivos del Problema de Investigación.

1.6. Hipótesis.

La implementación tanto del sistema de puntuación electrónico, así como del contador descendente permitirá contar con un moderno sistema de calificación para el jurado de la categoría de robot de batalla; y que el contador descendente, ayudará a que el juez del robot sumo pueda tener un mejor desempeño para validar los 5 segundos en los que el robot debe atacar después de su accionamiento inicial.

1.7. Metodología de Investigación.

La investigación en el lenguaje común se refiere a la búsqueda del conocimiento. También se puede definir la investigación como una la investigación científica y sistemática de información pertinente sobre un tema específico. La investigación, es una actividad académica y como tal, el término se debe utilizar en un sentido técnico.

Según (Kothari, 2004) en su texto que Clifford Woody, describe a la investigación como la comprensión de definir y redefinir los problemas, formular hipótesis o soluciones sugeridas; recolección, organización y evaluación de datos; y aplicar retenciones para llegar a conclusiones; y al fin

probar cuidadosamente las conclusiones para determinar si se ajustan a la formulación de hipótesis.

El propósito de la investigación es descubrir respuestas a preguntas mediante la aplicación de procedimientos científicos, aunque, cada estudio de investigación tiene su propio propósito específico. Los tipos básicos de investigación son los siguientes:

- a) La investigación descriptiva incluye encuestas y consultas de investigación de diferentes tipos. El propósito principal de la investigación descriptiva, es la descripción del estado de cosas tal y como existe en la actualidad.
- b) La investigación analítica, por otro lado, el investigador tiene que utilizar los datos o informaciones ya disponibles, y analizar estos para hacer una evaluación crítica del material.
- c) La investigación aplicada apunta a encontrar una solución para un problema inmediato que enfrenta una sociedad o una organización industrial/comercial.
- d) La investigación fundamental se refiere principalmente a las generalizaciones y con la formulación de una teoría.

CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA

2.1. Introducción a la microelectrónica

En lo que incumbe a microelectrónica estamos hablando de muchas grandes compañías que están día a día innovando, construyendo, creando y fabricando cada elemento que usamos siempre para crear cada aparato electrónico, aquí hago un breve recuento de las grandes compañías junto a su rango y porcentaje de fabricación a nivel mundial.

Tabla 2. 1: Ingresos por vendedores de microcontroladores.

Rango	Vendedor	Ingresos	Porcentaje
1	Intel	33.80	12.3
2	Samsung Electronics	20.46	7.5
3	Toshiba	11.82	4.3
4	Texas Instruments	11.77	4.3
5	Infineon + Qimonda	10.20	3.7
6	ST-Microelectronics	9.97	3.6
7	Hynix	9.10	3.3
8	Renesas	8.00	2.9
9	AMD	5.88	2.1
10	NXP	5.87	2.1
...	Otros	147.05	53.8
	TOTAL	237.91	100.00

Elaborado por: Gine Contreras Farah

2.1.1. Influencia económica de los semiconductores I

La Microelectrónica tiene un impacto mucho mayor en la economía mundial, sin embargo, debido a que está actuando como un controlador de tecnología para:

- Industria de la computación y el software
- Telecomunicaciones y medios de comunicación de la industria
- El comercio, la logística y el transporte
- Las ciencias naturales y la medicina
- Generación y distribución de energía
- Finanzas y administración

La Microelectrónica conduce la era de la información, es decir, que la microelectrónica tiene una enorme influencia económica como los avances que hay espuelas innovaciones en industrias y servicios. Si bien es cierto, la informática, las telecomunicaciones y productos de entretenimiento existían antes de la llegada de la microelectrónica. Finalmente, la sociedad de la información actual no habría sido posible sin la microelectrónica.

2.1.2. Impacto de los semiconductores en los bienes de consumo.

En la actualidad, los semiconductores se encuentran presentes en la mayoría de los bienes de consumo. Por ejemplo, los avances tecnológicos de los semiconductores en la telefonía fija y celular. En la telefonía celular tenemos a los teléfonos inteligentes y calculadoras científicas, tal como se muestra en la figura 2.1.



Figura 2. 1: Dispositivos electrónicos que utilizan semiconductores.
Fuente: (Manuel Arenas, 2014)

Lo que debemos saber acerca de las familias lógicas, es que es una colección de funciones parciales digitales que permiten:

- Montar a la lógica arbitraria , aritméticas y funciones de almacenamiento
- Son compatibles entre sí eléctricamente.
- Compartir una tecnología de fabricación común.

Tabla 2. 2: Listado de semiconductores.

Acrónimo	Significado
MOS	Metal Oxide Semiconductor.
FET	Field Effect Transistor (canal n- o p-).
BJT	Bipolar Junction Transistor (nnp or pnp).
CMOS	MOS complementario (circuito o tecnología).
CMOS Estático	Los datos almacenados en subcircuitos biestables y retenidos.
CMOS Dinámico	Los datos almacenados como cargas eléctricas se actualizan.
TTL	Transistor Transistor Logic (BJT y dispositivos pasivos).
ECL	Emitter-Coupled Logic (Lógica no saturante).
BiCMOS	CMOS y dispositivos bipolares en un solo chip.

Elaborado por: Gine Contreras Farah

El punto de vista de marketing, ¿Cómo puede funcionar y como el objetivo hace que se relacionen entre sí? De uso general IC. Normalmente los ejemplos son o muy simples o muy genéricos. Un circuito simple está compuesto por: compuertas (gates), flip -flops, counters, etc. Mientras, que por funcionalidad genérica tenemos: memorias RAM, ROM, microordenadores, FPL, etc.

2.1.3. Circuito integrado de aplicación específica (ASIC)

El productor estándar de aplicación específica (ASSP), diseñada para una tarea específica y es vendido a varios clientes. Ejemplos: los aceleradores gráficos, conjuntos de chips de radio celulares, chips de tarjetas inteligentes, etc. El circuito integrado para un usuario en específico (USIC), es diseñado y producido para una sola empresa. Ejemplos: procesador de audio para audífonos, etc.

2.1.4. Lógica de campo programable

No hay estructuras de diseño dedicado, no hay mascarillas dedicadas. La personalización es a través de medios puramente eléctricos. "Programable" es un nombre inapropiado ya que no hay secuencia de instrucción a ejecutar. "Configurable" es más precisa cuando se realizan subcircuitos prefabricados para formar el circuito objetivo. Todas las tecnologías de configuración actuales tienen sus raíces en la tecnología de memoria de semiconductores.

2.1.5. Síntesis de circuito automático III

La arquitectura o síntesis comienza a partir de un algoritmo de procesamiento de datos puramente de comportamiento. El código fuente no incluye indicaciones explícitas de cómo calcular las operaciones de procesamiento de datos y los recursos de hardware. Funciona en cinco fases principales:

- Identificar los requisitos computacionales y de almacenamiento.
- A partir de una biblioteca virtual, seleccione un bloque de construcción adecuados para cada tipo de procesamiento y almacenamiento operación.
- Establecer un calendario basado en el ciclo de la ejecución del algoritmo.
- Decidir sobre una organización de hardware capaz de ejecutar el plan de trabajo resultante.
- Hacer un seguimiento de los movimientos de datos y las operaciones para cada ciclo de reloj, se traducen en las instrucciones necesarias para la síntesis RTL.

2.1.6. Componentes virtuales

VC (también conocido como módulos de propiedad intelectual o núcleos) son paquetes de síntesis a disposición de los demás en una base comercial:

- Proveedor: Desarrolla una función importante en un modelo de síntesis para la venta.

- Licenciario: Compra el VC, incorpora en su diseño, lleva a cabo todo el resto, es decir, la síntesis, el lugar y la ruta (P & R), y la verificación general.
- VC son portables a través de las tecnologías de fabricación (módulos blandos), macro/megacells estándar que cumplen procesos específicos (módulos de disco duro).

La mayoría de los VC implementan funciones parciales bastante comunes, básicamente la parametrización pretende cubrir más aplicaciones.

2.1.7. Sistema de automatización de diseño electrónico de nivel

La presión hacia una mejor productividad de diseño ha incitado a la industria a buscar a la automatización del diseño desde una perspectiva más amplia.

- Metodología correcta por la construcción mediante el apoyo de refinamiento progresivo a partir de un prototipo virtual.
- Explorar el espacio de soluciones de arquitectura más sistemática y más rápida que con los métodos de síntesis RTL.
- Que sea posible para iniciar el desarrollo de software antes de completar el diseño de hardware.
- Mejorar la cobertura y la eficiencia de la verificación funcional al tratar con operaciones a nivel de sistema y mediante el aprovechamiento de la verificación formal.

2.1.8. Diseño a nivel de sistema

Las decisiones adoptadas en esta etapa determinan el resultado final sea más que cualquier otra cosa:

- Especificar la funcionalidad y características del sistema que sean.
- Se reparte la funcionalidad del sistema en subtarear.
- Explora compensaciones de hardware y software alternativos.
- Decidir sobre fabricación o compra para todos los principales bloques de construcción.
- Decidir sobre las interfaces y protocolos para el intercambio de datos.
- Decidir sobre formatos de datos, modos de operación, manejo de excepciones, etc.
- Definir , modelo , evaluar y refinar las diversas subtarear

2.1.9. El diseño de algoritmos

Agiliza los cálculos en vista de su implementación en hardware:

- Corta las exigencias de la carga y de la memoria de cálculo.
- Encontrar un compromiso entre la complejidad y precisión de cálculo.
- Contener los efectos debidos a finita cálculo de longitud de palabra.
- Decidir el número de esquemas de representación.
- Evaluar las alternativas y seleccionar la más adecuada.
- Cuantificar el mínimo de recursos computacionales requeridos.

2.2. Historia de los microcontroladores

- ... tubos de vacío, componentes discretos
- 1947 El primer transistor integrado (Bell Telephone Laboratories).
- 1959 El primer transistor bipolar plana.
- 1958 El primer circuito integrado disponible como un chip monolítico (flip-flop).
- 1965 El primer amplificador operacional.
- 1971: El primer microprocesador de 4 bits (Intel 4004).
- 1972: El primer microprocesador de 8 bits (Intel 8008).
- 1981: el primer PC de IBM.

2.2.1. Punto de Contacto de transistor de germanio



Figura 2. 2: Transistor que revolucionó las computadoras.
Fuente: (Nieto, 2015)

2.2.2. Invención del transistor

- el 23 de diciembre, 1947 (punto de contacto).
- el 30 de junio de 1948, la prensa era casi indiferente (bipolar).
- Los inventores del transistor son William Shockley, John Bardeen y Walter H. Brattain. Ellos recibieron el Premio Nobel de Física en 1956.
- Laboratorios Bell licencia libremente y difundió ampliamente en seminarios y papeles.
- En 1935, se concedió una patente a O. Heil para un triodo de efecto de campo.
- 1945: Bell Labs decidió limitar su investigación para el germanio y el silicio, los semiconductores más simples.

2.2.3. Invención Europea del transistor

Extraña y desconocida historia, reportado en Spectrum noviembre de 2005.

- Transistron, muy similar al transistor de Bell Labs, se inventó a finales de la Segunda Guerra Mundial en París, por dos científicos alemanes Herbert Mataré y Heinrich Welker.
- Se trabajó en Westinghouse, París.
- En 1948, una pequeña radio utiliza esta « transistron » (14 de mayo de 1948) Pero el gobierno francés y Westinghouse no pudo capitalizar " transistron " (la física nuclear más importante).

2.2.4. Transistron

Composición:

- Dos cables de metal.
- Se ponen en contacto plata germanio.
- Otro electrodo.
- Entra en contacto con la otra cara.

Un voltaje en este electrodo influye en la corriente a través de los otros.

Así se compuso un Transistor.



Figura 2. 3: Transistron
Fuente: (Wylie, 2010)

2.2.5. Comercialización del Transistor

- En 1958, el primer transistor de efecto campo estaba trabajando. Se le llamó "Tecnitron" por su creador, S. Tszner, que trabaja en Francia.
- Los ingenieros no les gustaba transistores; que preferían tubos. La primera demanda del mercado provino del mercado de audífonos, para los que la miniaturización era imprescindible.

- Sonotone en febrero de 1953; que contenía 5 transistores.
- A mediados de los años cincuenta, varias compañías estaban diseñando transistores - Raytheon, General Electric, Sylvania, RCA - y esto fue la producción en masa.
- Texas Instruments en 1953.

2.2.6. Silicon Valley- Fairchild

William Shockley, que dejó los Laboratorios Bell en 1954 para comenzar su propia compañía en Palo Alto, CA. - > Silicon Valley. Los jóvenes, tales como G. E. Moore y R. N. Noyce, se unieron a la empresa Shockley. Moore y Noyce, el "traidor", como Shockley vino a llamar a ellos, establecieron en 1957 Fairchild.

- Fairchild: en 1959 crea la nueva tecnología planar.
- Jack Kilby, ni Robert Noyce concibe el circuito integrado en 1959.

"Como el avance más significativo en el Instrumento de Texas desde... el transistor de silicio comercial".



Figura 2. 4: Científicos de Fairchild
Fuente: (Connie Guglielmo, 2013)

2.2.7. Transistor MOS

Incluso antes de Tszner en Francia se ha producido un cruce transistor de efecto de campo en 1958, muchos estudios estaban en curso en los EE.UU. sobre las posibilidades de un dispositivo de este tipo. En 1959, RCA estaba trabajando en FET. En 1962, RCA fabricaba bloque lógico de usos múltiples que comprende 16 MOS FET en un solo chip, sin embargo era extremadamente sensible a la carga estática, tensión de alimentación y efectos de óxido. Fairchild abandonó el proceso, incluso RCA cambió su énfasis de nuevo a bipolar.

A mediados de 1965, sólo dos empresas estaban produciendo MOS lcs, --- > P - MOS, NMOS, CMOS. En la figura 2.5 se muestra el esquemático interno de un transistor de tecnología MOS.

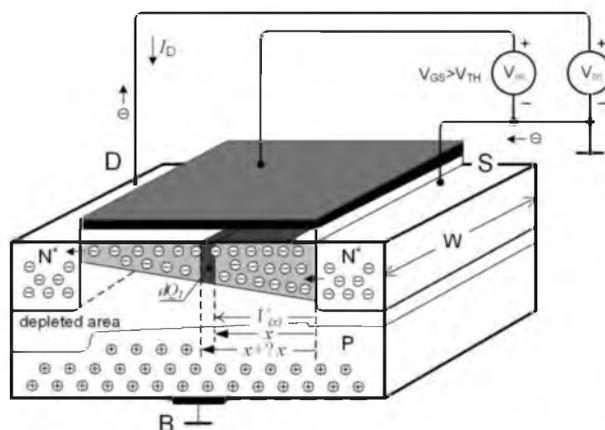


Figura 2. 5: Transistor MOS
Fuente: (Postreh, 2012)

2.3. Microcontroladores

Incluso en el momento en que Intel presenta el primer microprocesador, el 4004 ya había una demanda de microcontroladores: El contemporáneo microprocesador TMS1802, de Texas Instruments, diseñado para su uso en calculadoras, fue a finales de 1971 anunciado para aplicaciones en las cajas registradoras, relojes e instrumentos de medición.

El TMS 1000, que fue introducido en 1974, y este ya incluían memorias RAM, ROM, e I/O en el chip y puede ser visto como uno de los primeros microcontroladores, a pesar de que fue llamado un microordenador. Los primeros controladores para ganar uso muy extendido fuera el Intel 8048, que se integró en los teclados de PC, y su sucesor, el Intel 8051, así como la serie 68HCxx de microcontroladores de Motorola.

Hoy en día, los recuentos de producción de microcontroladores están en los miles de millones por año, y los controladores están integrados en muchos aparatos que ya nos hemos acostumbrado a utilizar muy a menudo, como:

- Electrodomésticos (microondas, lavadora, cafetera,...).
- Telecomunicaciones (teléfonos móviles).
- Industria del automóvil (de inyección de combustible, ABS,...).
- Industria aeroespacial.
- Automatización industrial.

Para (Guzmán M., 2011) un microcontrolador es un dispositivo que permite controlar diferentes aplicaciones de la Ingeniería. La importancia del uso de los microcontroladores, en los últimos años ha sido de gran utilidad para desarrollar tesis, trabajos de titulación o proyectos fin de carrera. Por ejemplo, (Madrid P., 2014) utilizó los microcontroladores para desarrollar tarjetas escudos utilizando el PIC18F4550.

2.3.1. Términos usados frecuentemente

Antes de que nos concentremos en microcontroladores, primero vamos a enumerar algunos términos que con frecuencia se encontrará en el campo de sistemas embebidos.

Microprocesador:

Se trata de una CPU normal (Central Processing Unit) como se puede encontrar en un PC. La comunicación con dispositivos externos se logra a través de un bus de datos, por lo tanto, el chip cuenta principalmente pines de datos y dirección, así como un par de pines de control. Todos los dispositivos periféricos (memoria, controlador de disco, controlador USB, contador de tiempo,...) se conectan al bus. Un microprocesador no puede ser operado de manera autónoma, al menos se requiere un poco de memoria y un dispositivo de salida para ser útil.

Tenga en cuenta que un procesador no es un controlador. Sin embargo, algunos fabricantes y proveedores de la lista da sus controladores bajo el

término "microprocesador". En este texto se utiliza el término procesador sólo por el núcleo del procesador (CPU) de un microcontrolador.

Microcontrolador:

Un microcontrolador ya contiene todos los componentes que le permiten operar independiente, y que ha sido diseñado en particular para el seguimiento de las tareas y / o de control. En consecuencia, además del procesador que incluye memoria, varios controladores de interfaz, uno o más temporizadores, un controlador de interrupciones, y por último pero no menos importante pines I/O que le permiten interactuar directamente con su entorno. Los Microcontroladores también incluyen operaciones de bits, que le permiten cambiar un bit dentro de un byte sin tocar los otros bits.

Controlador de Señal Mixta:

Este es un microcontrolador que puede procesar tanto señales digitales como analógicas.

Sistema Integrado:

Un área de aplicación importante para microcontroladores son sistemas embebidos. En los sistemas integrados, la unidad de control está integrada en el System3. A modo de ejemplo, pensar en un teléfono móvil, en el que el controlador está incluido en el dispositivo. Esto es fácilmente reconocible como un sistema embebido. Por otra parte, si se utiliza un PC normal en una

fábrica para controlar una cadena de montaje, esto también cumple muchas de las definiciones de un sistema embebido.

Sistema en tiempo real:

Los controladores se utilizan con frecuencia en sistemas de tiempo real, donde la reacción a un evento que tiene que ocurrir en un plazo determinado. Esto es cierto para muchas aplicaciones en la industria aeroespacial, ferrocarril, o áreas de automoción, por ejemplo, para el freno por cable en los coches.

Procesador incorporado:

Este término se produce a menudo en asociación con sistemas embebidos, y las diferencias a los controladores son a menudo muy borrosas. En general, el término "procesador embebido" se utiliza para dispositivos de gama alta (32 bits), mientras que el "controlador" se utiliza tradicionalmente para dispositivos de gama baja (4, 8, 16 bits). Motorola, por ejemplo, los archivos de sus controladores de 32 bits bajo el término "procesadores embebidos de 32 bits".

Procesador de señal digital (DSP):

Un área importante de su uso son las telecomunicaciones, por lo que su teléfono móvil es probable que contenga un DSP. Tales procesadores están diseñados para la adición y multiplicación rápida, que son las operaciones de teclas en el procesamiento de señales. Como las tareas que requieren un

procesador de señales también pueden incluir funciones de control, muchos proveedores ofrecen soluciones híbridas que combinan un controlador con un DSP en un chip, como DSP56800 de Motorola.

2.3.2. ¿Cuál es la diferencia entre un microprocesador y un microcontrolador?

Un microcontrolador suele incorporar otros componentes especializados que son útiles en los sistemas integrados de sistemas embebidos:

- Los puertos serie (RS - 232, USB)
- Establecimiento de una red (WiFi, Ethernet, ZigBee)
- La memoria de a bordo (Flash, DRAM, SRAM)
- Audio (MP3 codificación, procesamiento de señales)
- I/O analógico (DAC, ADC)

2.3.3. Familia de los microcontroladores

La mayoría de los fabricantes ofrecen una amplia gama de dispositivos de gama baja para aplicaciones de gama alta. La empresa Microchip envió 1 billón de microcontroladores en el otoño de 1999, 2 billones en la primavera de 2002 y 3 billones en invierno del 2004.

2.3.4. Fabricantes de microcontroladores

Existen muchos fabricantes de dispositivos microcontroladores, entre los más destacados en la enseñanza universitaria son los PICs de Microchip y Atmel (que utilizan los famosos Arduinos). A continuación, se muestra un listado de los fabricantes de microcontroladores a nivel mundial:

- Analog Devices
- Atmel
- Dallas Semiconductor
- Freescale Semiconductor
- Hitachi Semiconductor
- Intel
- Microchip
- National Semiconductor
- Renesas
- STMicro
- Texas Instruments
- Zilog

2.4. Microcontrolador de 8 bits

Los microcontroladores PIC tienen varias familias, entre las más básicas son los de 8 bits, tal como se muestra en la figura 2.6.



Figura 2. 6: Micro de 8 bits
Fuente: (Oswaldo Barajas, 2013)

En las siguientes sub secciones se describen brevemente la familia de los microcontroladores de 8 bits.

2.4.1. Diagrama de bloques de microprocesador de 8 bit

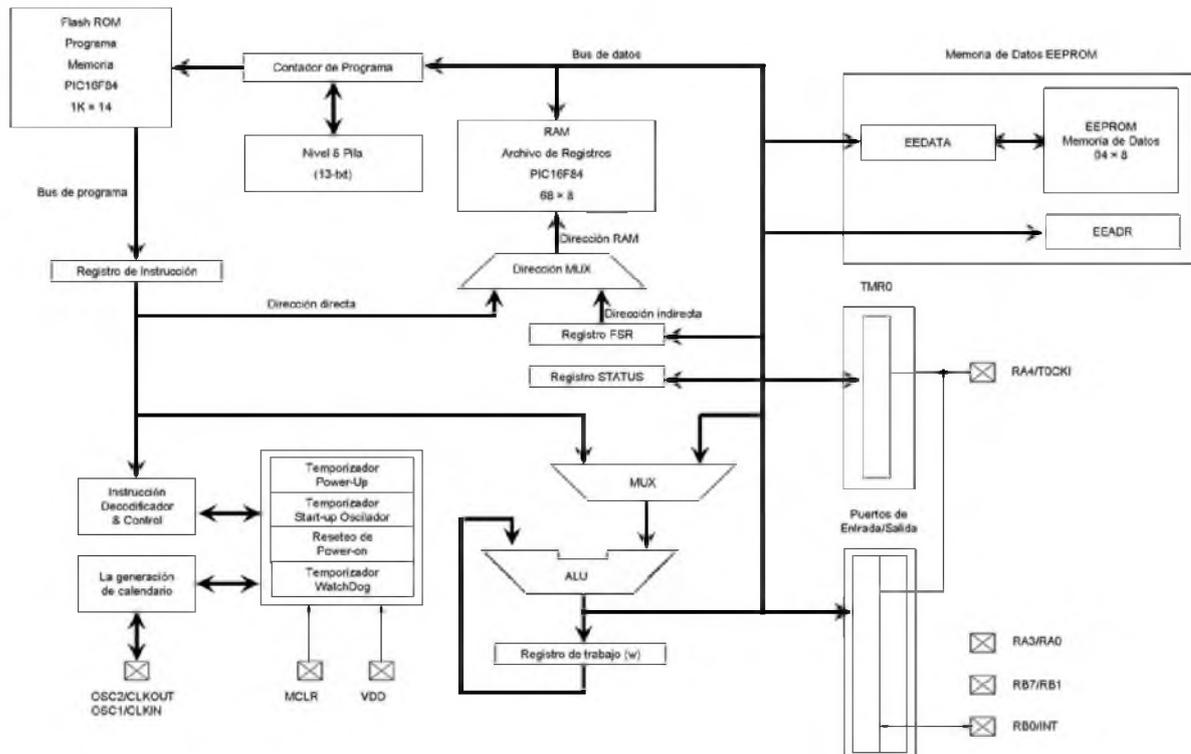


Figura 2. 7: Diagrama de bloques de un micro de 8 bits
Fuente: (Luis Galeano, Iván De Giacomi y Jabet Sánchez, 2008)

En la figura 2.7 mostramos el diagrama de bloques de un microcontrolador de 8 bits.

2.4.2. ¿Qué es un microcontrolador de 8 bits?

Un microcontrolador de 8 bits es un sistema autónomo de memoria, un procesador y los dispositivos periféricos que se puede utilizar como un sistema embebido. La mayoría de los microcontroladores de 8 bits programables que están en uso hoy en día están integrados en otros productos de maquinaria o de consumo, como teléfonos, automóviles, electrodomésticos, así como aparatos auxiliares de los sistemas informáticos.

Por lo tanto, otro nombre para un microcontrolador de 8 bit es "controlador incrustado de 8 bits". Algunos sistemas embebidos, son muy sofisticados, mientras que otras tienen requisitos mínimos para la memoria y la longitud de la programación con la complejidad del software bajo. Dispositivos de entrada y salida incluyen relays, solenoides, interruptores, pantallas LCD y sensores de datos, tales como la temperatura, el nivel de humedad o luz.

2.4.3. Tipos de microcontroladores de 8 bits

Hay muchos diferentes tipos de microcontroladores de 8 bits programables muchos de los tipos más comunes clasificados por tamaño Flash, tamaño de RAM, el número de líneas de entrada/salida, tipo de embalaje, la velocidad y la tensión de alimentación.

Los tamaños más comunes para la memoria RAM son 128 B, 192 B, 256 B, 368 B, 512 B, 768 B, 1 kB y 2 kB. También existen microcontroladores de 8 bits con memoria RAM de tamaños de hasta 768 kB. El tamaño de la Flash puede variar de 8 B a 4 MB, los tamaños más comunes son 1,75 kB, de 3,5 kb, 4 kB, 8 Kb, 16 Kb, 32 kB y 64 kB.

Los microcontroladores programables contienen pines de entrada/salida de propósito general, y su número puede variar. Los pines pueden ser configurados por el software a una entrada o un estado de salida. Cuando estos pasadores están configurados para un estado de entrada, que pueden ser utilizados para leer los sensores o las señales externas. Cuando están configurados para el estado de salida, estos pines pueden conducir los dispositivos externos tales como las pantallas LED y motores.

2.4.4. Aplicaciones para los microcontroladores de 8 bits:

Los microcontroladores programables de 8 bits están diseñados para ser utilizados para aplicaciones embebidas, al contrario de los microprocesadores que se pueden encontrar en los PC. Los microcontroladores de 8 bits se utilizan en dispositivos controlados automáticamente incluyendo los dispositivos médicos implantables, herramientas eléctricas, juguetes, máquinas de oficina, sistemas de control de motor, controles remotos, electrodomésticos, etc.

2.5. Microcontrolador de 16 bits

En la figura 2.8 se muestra el encapsulado de un microcontrolador de 16 bits a 32 bits cuya familia es XC2000.



Figura 2. 8: Micro de 16/32 bit
Fuente: (Oswaldo Barajas, 2013)

2.5.1. Diagrama de bloques de un microcontrolador de 16 bits

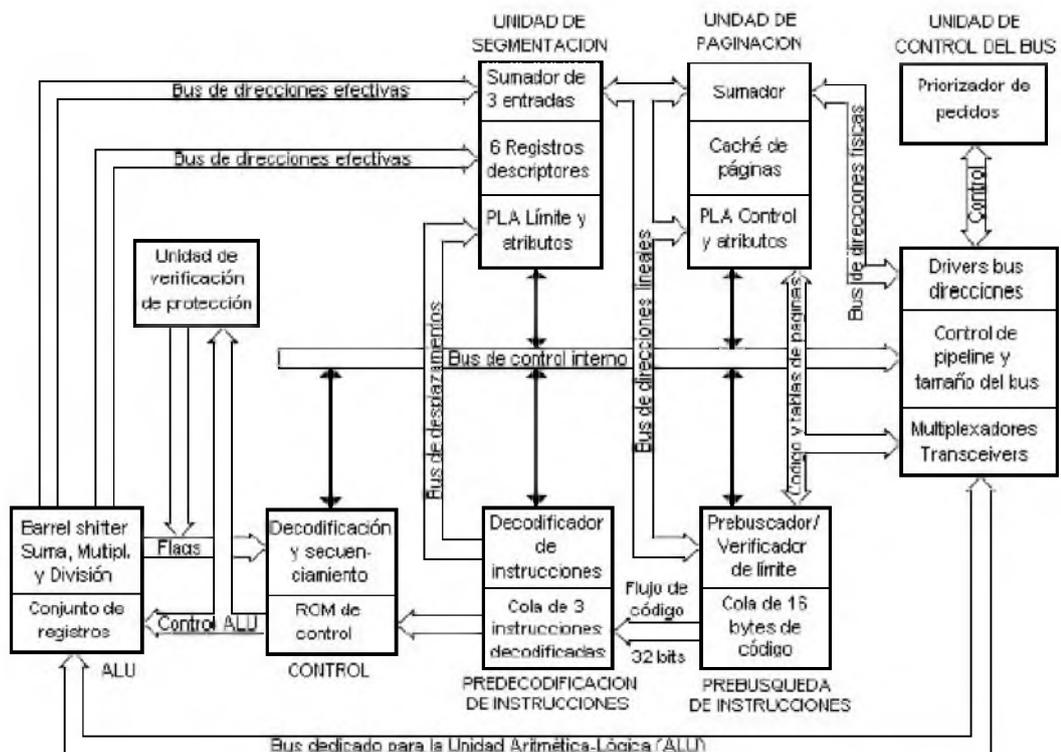


Figura 2. 9: Diagrama de bloques de un micro de 16 bits
Fuente: (Nacho, 2009)

En la figura 2.9 vemos el diagrama de bloques de un microcontrolador de 16 bits.

2.5.2. ¿Qué es un microcontrolador de 16 bits?

Un microcontrolador de 16 bits es un sistema autónomo de memoria, periféricos y un procesador que se puede utilizar como un sistema embebido. La mayoría de los microcontroladores de 16 bits programables utilizados en la actualidad están integrados en otros productos de consumo o maquinaria, como teléfonos, automóviles, electrodomésticos y periféricos para los sistemas informáticos.

Por lo tanto, otro nombre para un microcontrolador de 16 bits está "incrustado controlador de 16 bits. "Algunos sistemas embebidos son más sofisticados, mientras que otras tienen requisitos mínimos para la memoria y la longitud de la programación con una baja complejidad para el software. Los dispositivos de entrada y salida incluyen pantallas LCD, relays, solenoides, interruptores y sensores de datos, tales como la temperatura, la humedad o el nivel de luz, entre otros.

2.5.3. Tipos de Microcontroladores de 16 bits

Hay muchos tipos de microcontroladores programables de 16 bits. Normalmente los tipos más comunes son clasificados por varios parámetros que incluyen el tamaño del flash, el número de líneas de entrada/salida, tipo de embalaje, tamaño de RAM, tensión de alimentación y la velocidad.

Los tamaños más comunes para la memoria RAM son de 1,5 kb, 2 kB, 4 kB, 8 kB, 16 kB y 32 kB. También llevamos 16 bits microcontroladores con memoria RAM de tamaños de hasta 768 kB. Los tamaños de la Flash pueden variar de 4 kB de 1,5 MB, con los tamaños más comunes son 16 kB, 32 kB, 64 kB, 128 kB, 256 kB y 512 kB.

Los microcontroladores programables contienen pines de E/S de propósito general y el número de estos pines pueden variar en función del microcontrolador. Pueden ser configurados por el software a un estado de entrada o de salida. Cuando estos pines están configurados para un estado de entrada, pueden ser utilizados para leer las señales externas o sensores. Cuando están configurados para el estado de salida, estos pines pueden conducir dispositivos externos, como motores y pantallas LED.

2.5.4. Aplicaciones para microcontroladores de 16 bits

Los microcontroladores programables de 16 bits están diseñados para ser utilizados en aplicaciones embebidas, al contrario de los microprocesadores que se pueden encontrar en los PC. Los microcontroladores de 16 bits se utilizan en dispositivos controlados automáticamente incluyendo los dispositivos médicos implantables, herramientas eléctricas, juguetes, máquinas de oficina, sistemas de control de motor, controles remotos, electrodomésticos, así como otros tipos de sistemas embebidos.

2.5.5. Microcontrolador de 32 bits

En la figura 2.8 se muestra el encapsulado de un microcontrolador de 16 bits a 32 bits cuya familia es XC2000.



Figura 2. 10: Micro de 16/32 bit
Fuente: (Oswaldo Barajas, 2013)

2.5.6. Diagrama de bloque de un microcontrolador de 32 bits.

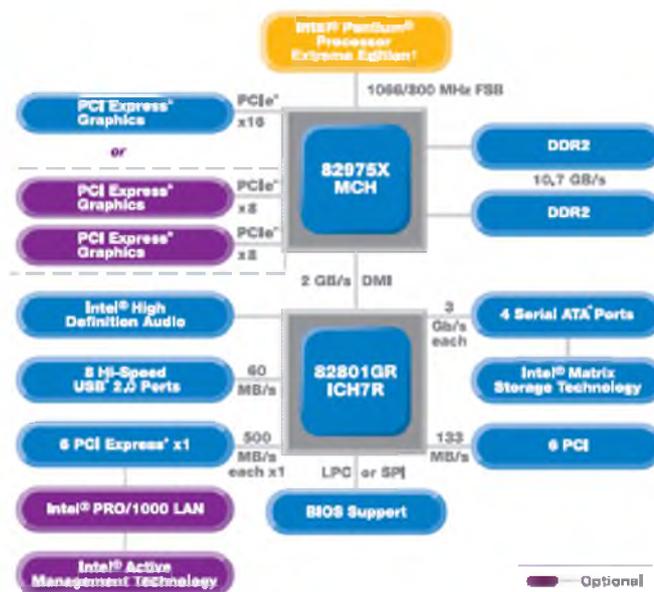


Figura 2. 11: Micro de 16/32 bit
Fuente: (Hector Hernando, 2010)

En la figura 2.11 vemos el diagrama de bloques de un microcontrolador Intel Pentium 4.

2.5.7. ¿Qué es un microcontrolador de 32 bits?

Un microcontrolador 32 bits es una pequeña computadora en un solo circuito integrado que contiene la memoria, un núcleo de procesador, y los periféricos de entrada / salida programables. La memoria de programa tales como OTP ROM o NOR flash y una pequeña cantidad de RAM también se puede encontrar en el chip.

2.5.8. Tipos de microcontroladores de 32 bits

Hay muchos diferentes tipos de microcontroladores de 32 bits los tipos de microcontroladores de 32 bits más comunes son clasificados por tamaño de RAM, el tamaño de Flash, el número de líneas de entrada / salida, velocidad, tensión de alimentación y el tipo de embalaje. Nuestros filtros paramétricos ayudan a restringir los resultados de búsqueda en función de las especificaciones requeridas.

Los tamaños más comunes para la memoria RAM son 4 kB, 8 KB, 16 KB, 32 KB, 48 KB, 64 kB y 128 kB. También llevamos tamaños de hasta 1 MB. Los tamaños de la Flash pueden variar de 0 a 4 MB B, con los tamaños más comunes son 32 kB, 64 kB, 128 kB, 256 kB, 512 kb y 1 MB.

Los microcontroladores pueden contener un número variable de pines de E/S de propósito general. Los pines pueden ser configurados por el software de entrada o salida de un estado. Cuando estos pasadores están configurados para un estado de entrada, que pueden ser utilizados para leer

los sensores o las señales externas. Cuando están configurados para el estado de salida, estos pueden conducir dispositivos externos como motores o LEDs.

2.5.9. Aplicaciones para microcontroladores de 32 bits

Los microcontroladores programables de 32 bits están diseñados para ser utilizados en aplicaciones embebidas, al contrario de los microprocesadores que se pueden encontrar en los PC. Los microcontroladores de 32 bits se utilizan en dispositivos controlados automáticamente incluyendo los dispositivos médicos implantables, herramientas eléctricas, juguetes, máquinas de oficina, sistemas de control de motor, controles remotos, electrodomésticos, así como otros tipos de sistemas embebidos.

2.6. Microcontrolador PIC 18F4550

2.6.1. Diagramas de Pines.

En las figuras 2.12 y 2.13 se muestran los esquemáticos del microcontrolador PIC18F4550 para 40 y 44 pines, respectivamente. Cada figura se muestra la descripción de los pines de entrada y salida, así como los pines para conectar el oscilador. Para (Bruno S., 2015) una de las ventajas que tiene el PIC18F4550 es que requiere solo de 2 pines para ser programado, ya sea utilizando lenguajes de programación de bajo y alto nivel. Aunque lo recomendable, es utilizar compiladores de alto nivel.

40-Pin PDIP

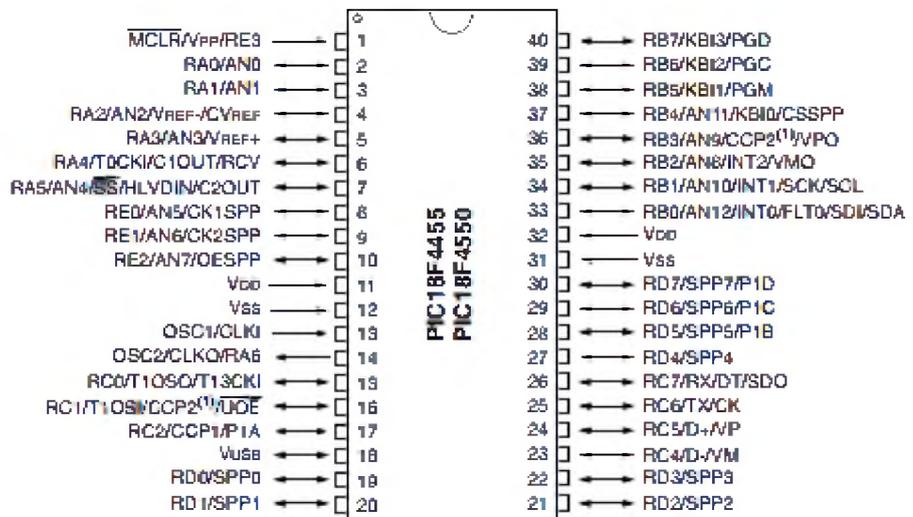


Figura 2. 12: Diagrama esquemático del PIC 18F4550 de 40 pines.
Fuente: (Microchip, 2006)

44-Pin TQFP

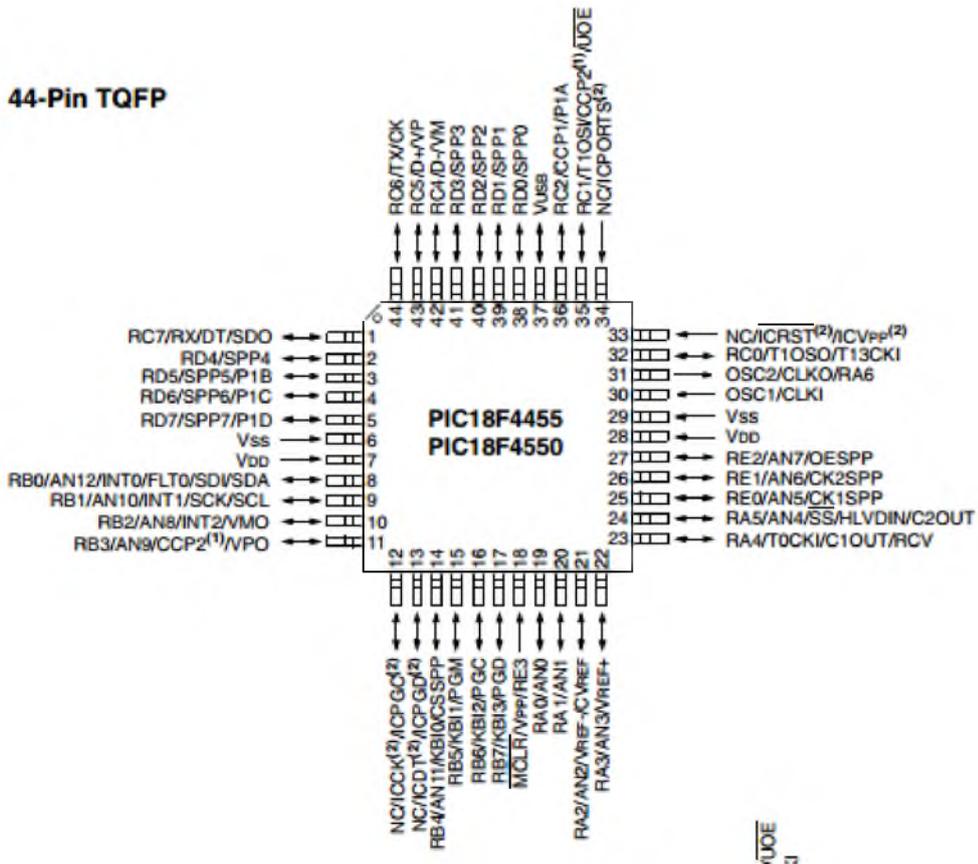


Figura 2. 13: Diagrama esquemático del PIC 18F4550 de 44 pines.
Fuente: (Microchip, 2006)

A continuación, en la figura 2.14 se muestra la arquitectura interna del microcontrolador PIC18F4550.

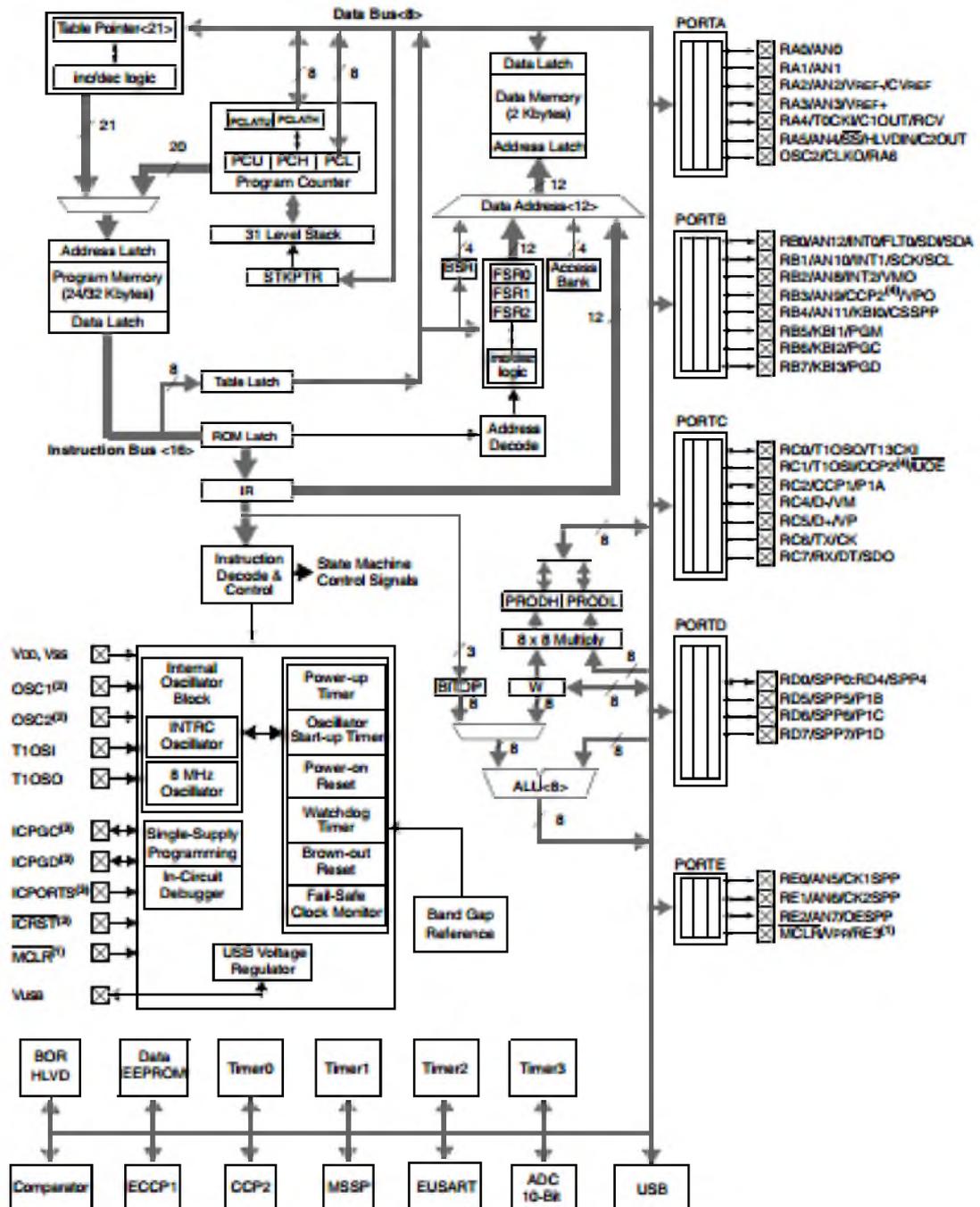


Figura 2. 14: Diagrama de PIC 18f4550
Fuente: (Microchip, 2006)

CAPÍTULO 3: SIMULACION Y RESULTADOS OBTENIDOS

3.1. Diseño del Sistema de Puntuación Electrónico.

El propósito de diseñar un sistema de puntuación electrónica, surgió del Concurso Ecuatoriano de Robótica. Específicamente, en la categoría de robots de batalla, los jueces califican de acuerdo al ataque con armas, embestidas, vuelcos e inmovilizaciones, y de acuerdo a reglamentos, estos tienen una puntuación determinada por los organizadores de los concursos de robótica.

Este tipo de puntuaciones son llevados manualmente, en una hoja entregada por los organizadores. El trabajo de titulación, propone un sistema electrónico que se comunica inalámbrica a una PC para que se muestre en pantallas gigantes, los resultados en tiempo real de las batallas de 120 libras y 30 libras.

A continuación, se describen los apartados de la parte del diseño de puntuación electrónica, así como el diseño PCB del sistema y los resultados obtenidos en las simulaciones sobre el software Proteus.

3.2.1. Diseño del Sistema de Puntuación Electrónico.

El circuito de la figura 3.1 muestra el diseño esquemático del sistema de puntuación electrónica desarrollado en el software Proteus 7.9.

encargado de controlar todas las tareas previamente programadas en CCS C Compiler.

Adicionalmente, podemos ver en la figura 3,1 que el PIC 18F4550, dispone de pines de E/S, que para este proyecto consta de 6 botoneras que son: A, B, C, D, E y F que se activan con bajo y tienen resistencias de pull up de 10k Ω . Los pines del puerto A, existe un dipswitch de 4 bits para poder configurar la dirección del dispositivo electrónico diseñado en la red inalámbrica mediante un Zigbee, es decir, que desde "0000" hasta "1111" se podría configurar hasta 16 equipos de marcación electrónica.

Ahora, mediante 3 diodos led se verifica la correcta pulsación de las botoneras para las puntuaciones para 5, 10 y 20 puntos, que son embestidas, uso de armas e inmovilizaciones respectivamente.

Adicionalmente, se utiliza el dispositivo inalámbrico Zigbee (XBee) que es el encargado de entablar la comunicación inalámbrica entre el dispositivo diseñado y un equipo remoto, que se encuentra conectado a la computadora de escritorio o una laptop.

Para alimentar al módulo Zigbee, así como ocurrió en la etapa de reducción de voltaje 5 Vdc para alimentar al PIC 18F4550, este módulo tiene que alimentarse con un voltaje de polarización de 3.3 Vdc. Por lo tanto, se tuvo que colocar 2 diodos 1N4007 en serie para reducir el voltaje a 3.3 Vdc.

Adicionalmente, se realizó un circuito divisor de voltaje para reducir los 5 Vdc que provienen del microcontrolador, y así mismo otro circuito mediante 2 transistores y 3 resistores para elevar los 3.3 V que vienen del Zigbee a 5 Vdc para que entren en el microcontrolador PIC 18F4550.

3.2.2. Programación de alto nivel en CCS C Compiler para el sistema de marcación electrónica.

Para nuestro trabajo de titulación el código que se describe a continuación en las siguientes figuras corresponde a como fue estructurado siguiendo el orden lógico del diagrama de flujo para su aplicación. Este un contador que va a acumulando en los registros internos los valores que correspondientes a cinco, diez o veinte de manera incremental y decrece cinco dependiendo mucho del transcurso de la competencia y las circunstancias que se presenten.

En la siguiente Figura 3. 2, la línea 1 describe el llamado de la librería y la configuración previa para el funcionamiento de nuestro PIC 18f4550 podemos observarlo en las líneas 2, 3, 4.

Para “#FUSES HS” nos refiere a la velocidad el cual funcionara el microcontrolador, en la línea 3 nos detalla la velocidad del reloj y en la línea 4 indicamos el protocolo de comunicación con su respectiva velocidad en baudios y los pines para transmisión y recepción.

En las siguientes líneas describen una asignación de puertos A, B, C, D y E del uC, pudiendo utilizar simplemente la palabra “porta” por ejemplo en un momento determinado de la función principal.

```
UC-1Total.c*
1  #include<18f4550.h> //micropic
2  #FUSES HS
3  #use delay (clock=20000000) // o 4Mhz
4  #use rs232(baud=9600, xmit=pin_c6,rcv=pin_c7)
5  #byte porta=0xf80
6  #byte portb=0xf81
7  #byte portc=0xf82
8  #byte portd=0xf83
9  #byte porte=0xf84
10
```

Figura 3. 2: Descripción del encabezado del programa.
Elaborado por: Gine Contreras Farah

A continuación en la Figura 3. 3 en las líneas 11 hasta el 15 se declaran las variables globales que se utilizaran en nuestro programa y en su línea 9 la sentencia correspondiente para el proceso de interrupciones por comunicaciones que serán de vital importancia para el intercambio de datos con el PC.

```
11  int x;
12  int id;
13  int16 puntaje=0;
14  int valor[6];
15  #int_RDA
```

Figura 3. 3: Declaración de variables globales
Elaborado por: Gine Contreras Farah

La función RDA_isr(), es la principal responsable para la interrupción que ocurre en la comunicación serial con el protocolo de comunicación RS-232, este responde a su llamado en determinado tiempo y conjuntamente con una sentencia “IF” para comparar el valor del arreglo de la variable “valor” si es igual a la identificación del control al que representa este microcontrolador.

Hay que tener en cuenta que el arreglo “valor[2]” se compara con la variable “id” ya que más adelante en la función principal del programa es asignado como el control 1 o 2 dependiendo de los pines activados en el dipswitch construido para su efecto, así que este programa está destinado para cualquiera de los dos uC encargados de hacer el puntaje.

Una vez identificado el dispositivo, se encarga de transmitir el valor que posee en el registro de la variable “puntaje” al PC.

```
16
17 void RDA_isr()
18 {
19     gets(valor);
20     if(valor[0]=='U' && valor[1]=='C' && valor[2]==id)
21     {
22         printf("UC1:%ld%c",puntaje,13);
23         delay_ms(500);
24     }
25
26 }
27
```

Figura 3. 4: Función void RDA_isr para identificar el dispositivo y envío de datos acumulados en los registros internos.
Elaborado por: Gine Contreras Farah

El Juez designado en la competencia de batalla de robot tendrá que ir calificando por medio de nuestro sistema de puntuación electrónica, para ello en la interfaz construida tenemos varios pulsadores y cada uno aumenta o decrece los valores según las circunstancias que se presenten en dicha competición. La función presentada a continuación en la figura 3.5 representa al incremento de cinco puntos que se realiza pulsando el pin asignado “PIN_D0”, mediante una instrucción “while” se incrementa de cinco en cinco y

manteniendo guardada el valor de esta variable para su posterior envío al PC cuando se lo requiera.

```
28 void incrementa_cinco()
29 {
30     while(!input(PIN_D0));
31     puntaje=puntaje+5;
32     for(x=0; x<= 80; x++)
33     {
34         delay_ms(1);
35         output_high(PIN_B1);
36         output_high(PIN_B0);
37     }
38     output_low(PIN_B1);
39     output_low(PIN_B0);
40 }
41
```

Figura 3. 5: Función incrementa_cinco, encargada de aumentar el valor de la variable unidad y decena según las circunstancias de la competencia.

Elaborado por: Gine Contreras Farah

El dispositivo está diseñado no solo para incrementar valores de cinco, sino que también tiene pines para incrementar diez y veinte puntos como se muestra en las figuras 3.6 y 3.7. Para su perfecto funcionamiento de estas funciones se asignó un pin específico que cumpla con lo requerido, en el caso de 10 puntos se declaró al pin D1 y para 20 el pin D2.

```
42 void incrementa_diez()
43 {
44     while(!input(PIN_D1));
45     puntaje=puntaje+10;
46     for(x=0; x<= 80; x++)
47     {
48         delay_ms(1);
49         output_high(PIN_B1);
50         output_high(PIN_B0);
51     }
52     output_low(PIN_B1);
53     output_low(PIN_B0);
54 }
55
```

Figura 3. 6: Función incrementa_diez para el incremento para diez puntos

Elaborado por: Gine Contreras Farah

```

56 void incrementa_veinte()
57 {
58     while(!input(PIN_D2));
59     puntaje=puntaje+20;
60     for(x=0; x<= 80; x++)
61     {
62         delay_ms(1);
63         output_high(PIN_B1);
64         output_high(PIN_B0);
65     }
66     output_low(PIN_B1);
67     output_low(PIN_B0);
68 }
69

```

Figura 3. 7: Función incrementa_veinte para el incremento para veinte puntos.
Elaborado por: Gine Contreras Farah

Así mismo cuando se restan puntos en la competencia este debe de decrecer los valores de la variable puntaje que es responsable de tener los datos en su registro interno, para ello creamos la función “decrementa” descrito en la figura 3.8.

Dentro de la sentencia “while” tenemos la operación específica que tendrá la labor de descontar 5 puntos al valor que posee la variable.

```

70 void decrementa()
71 {
72     while(!input(PIN_D5));
73     puntaje=puntaje-5;
74     for(x=0; x<= 80; x++)
75     delay_ms(1);
76
77 }
78

```

Figura 3. 8: Función decrementa, encargada de descontar 5 puntos a los valores que posee actualmente la variable “puntaje”.
Elaborado por: Gine Contreras Farah

Por último es necesario la creación de la función void main para poder ejecutar el programa dentro de nuestro microcontrolador.

Las primeras líneas de nuestra función principal hacen referencia a las entradas y salidas designadas por hexadecimal en cada puerto previamente descrito en el inicio del código y su inicialización en cero para cada uno.

Adicionalmente se ha realizado una operación matemática sencilla que aumenta en 48 el valor de la variable `id`, esto se debe a que la transmisión se da por código ASCII, y el número 1 o 2 corresponden al valor 49 o 50 respectivamente.

A continuación describimos la habilitación de las interrupciones globales de la comunicación serial con el protocolo RS-232, para que este pueda pausar el programa en cualquier instante de tiempo sin importar su tarea que este efectuando en ese momento.

Conseguimos realizar un parpadeo de los leds indicadores en los pines B0 y B1 con las instrucciones `output_high` y `output_low` para indicarnos que el programa inicio sin novedad alguna. Con la sentencia `while (true)`, lo que se propone es entrar a un ciclo sin fin para comparar las entradas de los pines D0, D1, D2, Y D5 ya que estos representan a los pulsadores que cuentan los puntos a favor o en contra en la competencia.

```

79 void main(void)
80 {
81
82     set_tris_a(0x0f);
83     set_tris_b(0x00);
84     set_tris_c(0x80);
85     set_tris_d(0x3f);
86     set_tris_e(0x00);
87     porta=0;
88     portb=0;
89     portc=0;
90     portd=0;
91     porte=0;
92     id=porta+48;
93
94     enable_interrupts(INT_RDA);
95     enable_interrupts(GLOBAL);
96     delay_ms(100);
97     output_high(PIN_B0);
98     output_high(PIN_B1);
99     delay_ms(300);
100    output_low(PIN_B1);
101    output_low(PIN_B0);
102    while(true)
103    {
104        if(!input(PIN_D0))
105            incrementa_cinco();
106        if(!input(PIN_D1))
107            incrementa_diez();
108        if(!input(PIN_D2))
109            incrementa_veinte();
110        if(!input(PIN_D5))
111            decrementa();
112    }
113 }

```

Figura 3. 9: Función principal del código para el sistema de puntuación.
Elaborado por: Gine Contreras Farah

Para efecto de simulación de nuestros dispositivos, el algoritmo está diseñado adecuadamente y su correspondiente demostración en la presentación de este trabajo de titulación fue necesario realizarlo con un código previo y sencillo que se muestra en la figura 3.10. Como se puede apreciar el código descrito incluye la función de interrupción por comunicaciones, y únicamente tres funciones realizan la tarea de incrementar unidad, decena y decremento, además de la encargada en el reconocimiento

del dispositivo y su función principal. El funcionamiento del mismo es sencillo y de muy fácil comprensión para el público en general.

```

1  #include<18f4550.h> //micropic
2  #FUSES HS
3  #use delay (clock=2000000) // o 4Mhz
4  #use rs232(baud=9600, xmit=pin_c6,rcv=pin_c7)
5  int unidad=0;
6  int decena=0;
7  int valor[4];
8  #int_RDA
9  void RDA_isr()
10 {
11     gets(valor);
12     if(valor[0]=='U' && valor[1]=='C' && valor[2]=='1')
13     {
14         printf("UC1%d%d%c",decena,unidad,13);
15     }
16 }
17 void incrementa_local()
18 {
19     while(!input(PIN_D0));
20     unidad++;
21     if(unidad>9)
22     {
23         unidad=0;
24         decena++;
25         if(decena>9)
26             decena=0;
27     }
28 }
29 void decrementa_local()
30 {
31     while(!input(PIN_D1));
32     if(unidad!=0 || decena!=0)
33     {
34         if(unidad==0)
35         {
36             unidad=9;
37             decena--;
38         }
39         else{
40             unidad--;}
41     }
42 }
43 void main(void)
44 {
45     enable_interrupts(INT_RDA);
46     enable_interrupts(GLOBAL);
47     while(true)
48     {
49         if(!input(PIN_D0))
50             incrementa_local();
51         if(!input(PIN_D1))
52             decrementa_local();
53     }
54 }

```

Figura 3. 10: Código para efecto de simulación y demostración del sistema de puntuación.

Elaborado por: Gine Contreras Farah

3.2.3. Resultados obtenidos del sistema de marcación electrónica en Proteus.

En la sección anterior se demostró la simulación del sistema de puntuación electrónica para realizar el diseño de PCB en Ares de Proteus. Para esta sección, se demuestra los resultados obtenidos una vez que se ha ejecutado la simulación. El propósito, es verificar que el algoritmo utilizado en la sección 3.2.2 sea el más apropiado. En la figura 3.11 se muestra el diseño del circuito de puntuación electrónica diseñado en Proteus.

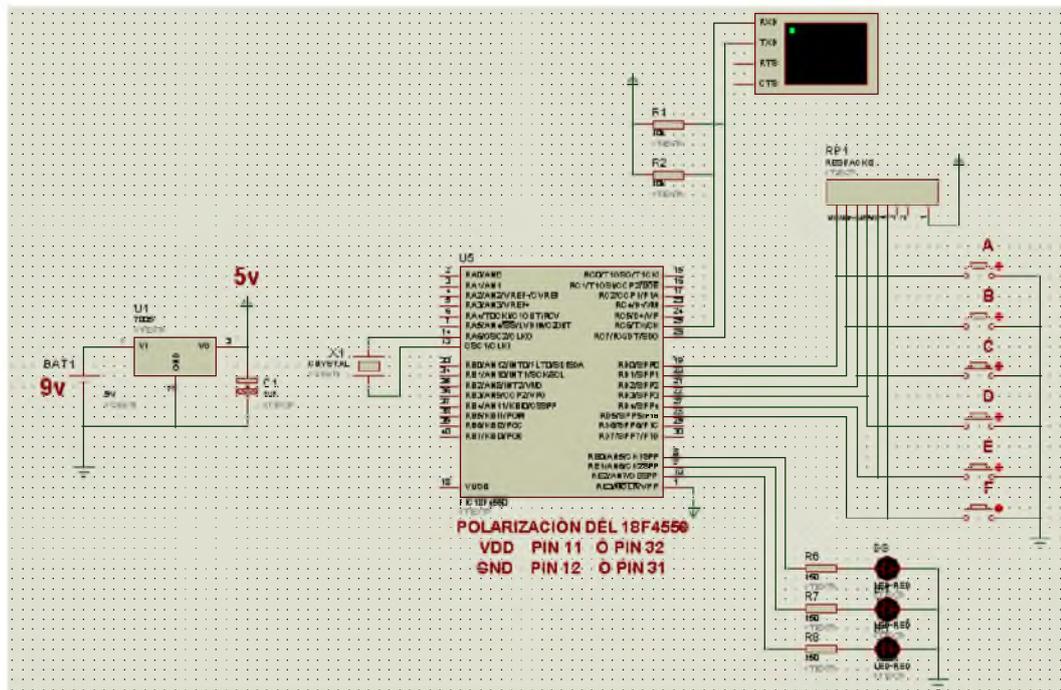


Figura 3. 11: Circuito para verificar la simulación del equipo de marcación de puntos para la categoría de robot de batalla.
 Elaborado por: Gine Contreras Farah

Una vez que se ejecuta la simulación del sistema de puntuación, se abre una ventana conocida como Hyper Terminal. En la figura 3.12 se muestra el resultado obtenido al presionar la botonera “A”, la cual indica una puntuación de 20 puntos por inmovilizar al oponente por ataque.



Figura 3. 12: Resultado obtenido para inmovilizar al oponente por ataque.
Elaborado por: Gine Contreras Farah

En la figura 3.13 se muestra el resultado obtenido al presionar la botonera “B”, la cual indica una puntuación de 5 puntos por embestidas de un robot de batalla a su oponente.

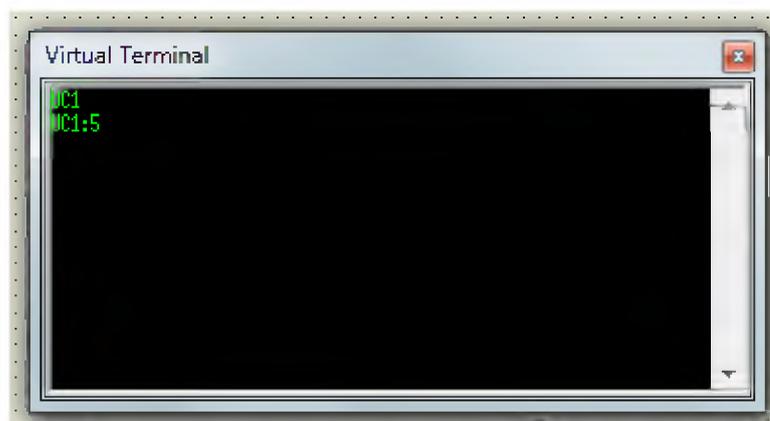


Figura 3. 13: Resultado obtenido para embestidas.
Elaborado por: Gine Contreras Farah

En la figura 3.14 se muestra el resultado obtenido al presionar la botonera “B”, la cual indica una puntuación de 10 puntos por vuelcos de un robot de batalla.

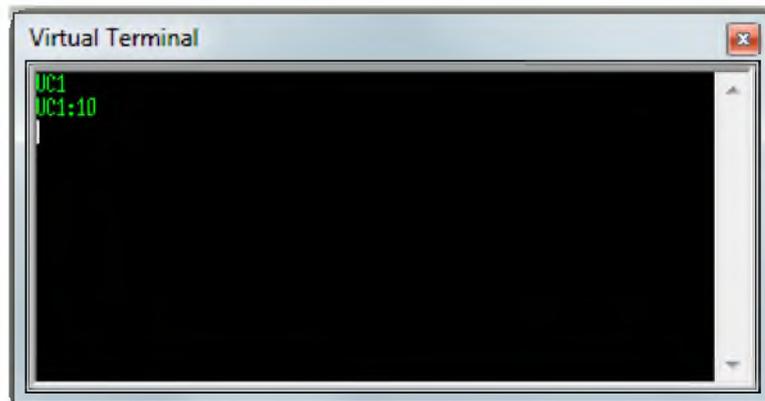


Figura 3. 14: Resultado obtenido para vuelcos.
Elaborado por: Gine Contreras Farah

3.2.4. Diseño electrónico PCB del sistema de marcación electrónica en Proteus.

Para una aplicación profesional de nuestros dispositivos fue necesario el diseño electrónico PCB en el programa ARES para su presentación ante el Concurso Ecuatoriano de Robótica CER 2015 realizado en nuestra magna institución organizado por nuestra facultad.

Como se podrá apreciar en las siguientes figuras, el proceso final de nuestros diseños se llegó a terminar con un acabado profesional.

Por un lado tenemos el diseño de las botoneras y consiguiente el sistema de comunicación con el microprocesador correspondiente aplicado en el mismo.

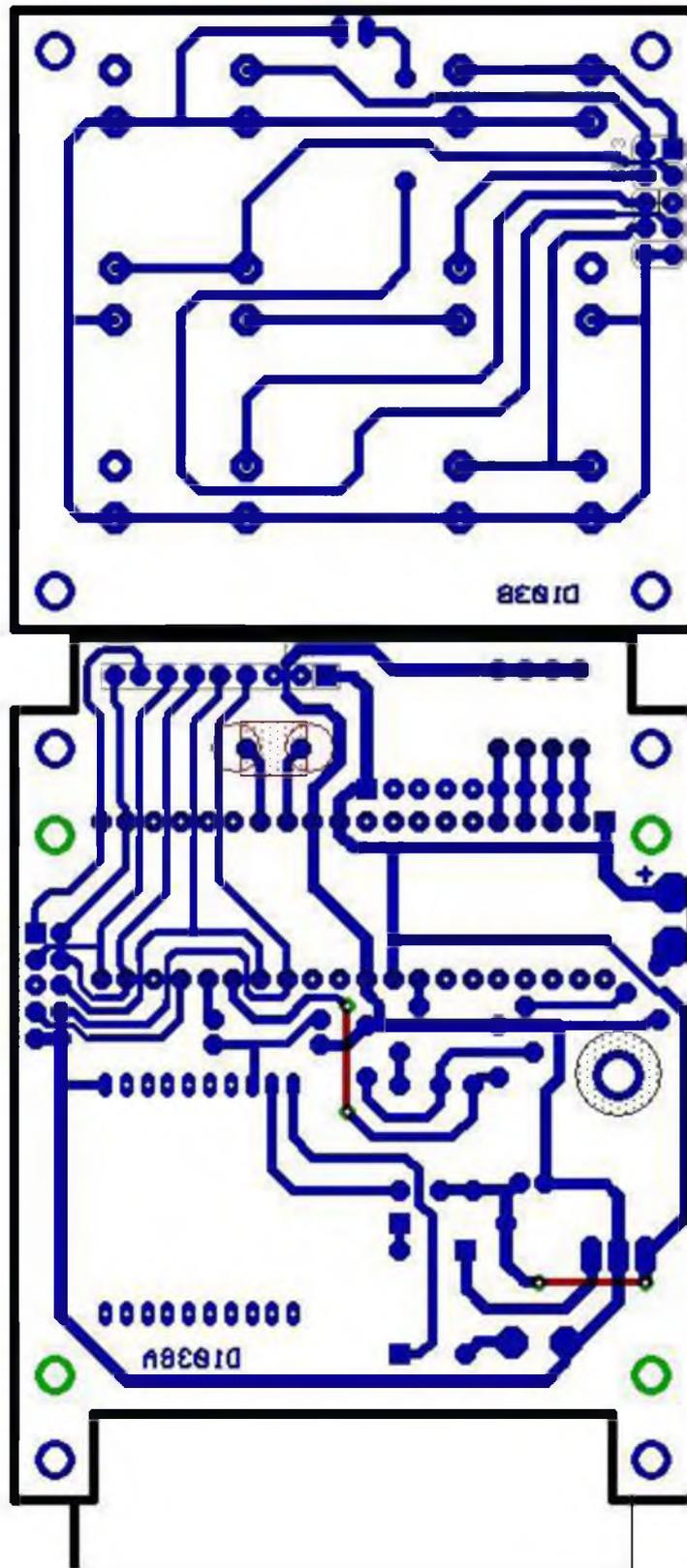


Figura 3. 15: Vista superior del PCB para el sistema de puntuación.
Elaborado por: Gine Contreras Farah

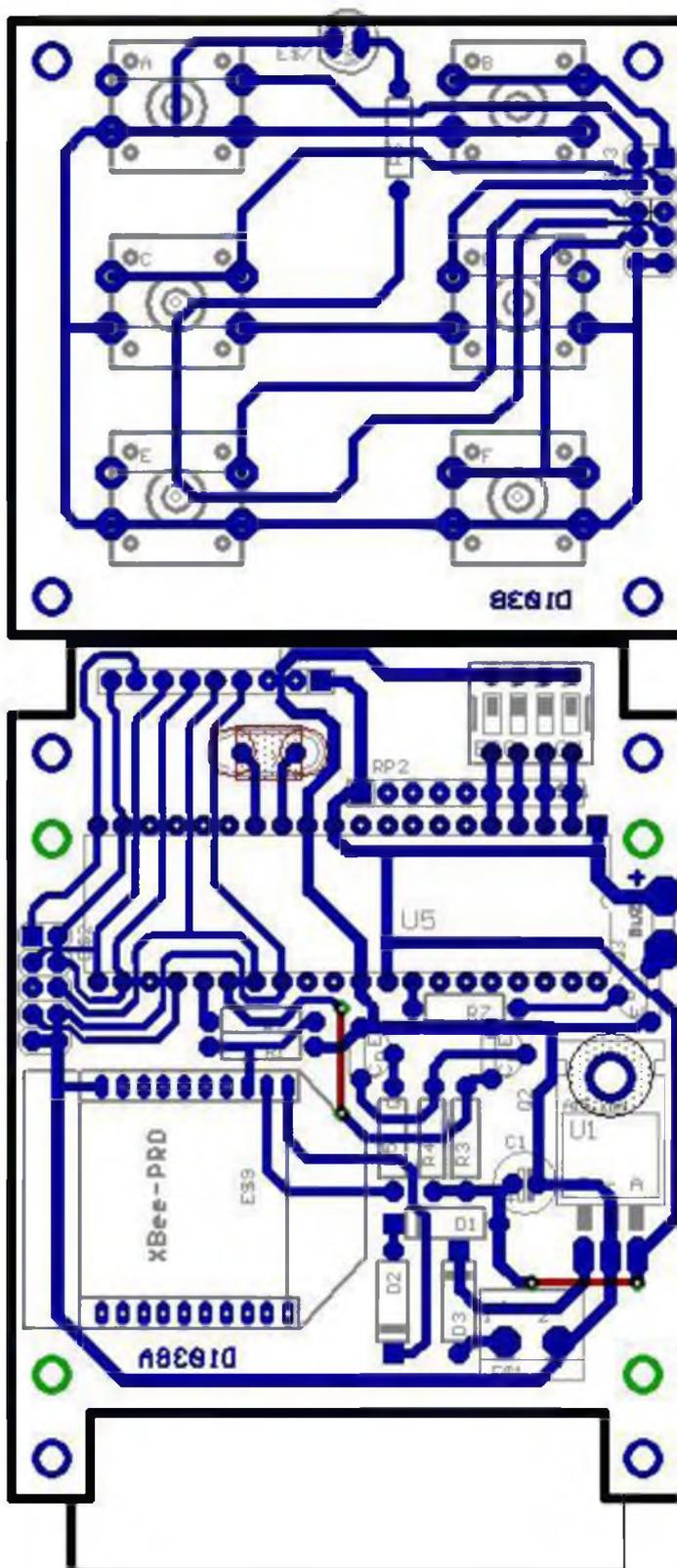


Figura 3. 16: Vista inferior del PCB para el sistema de puntuación.
 Elaborado por: Gine Contreras Farah

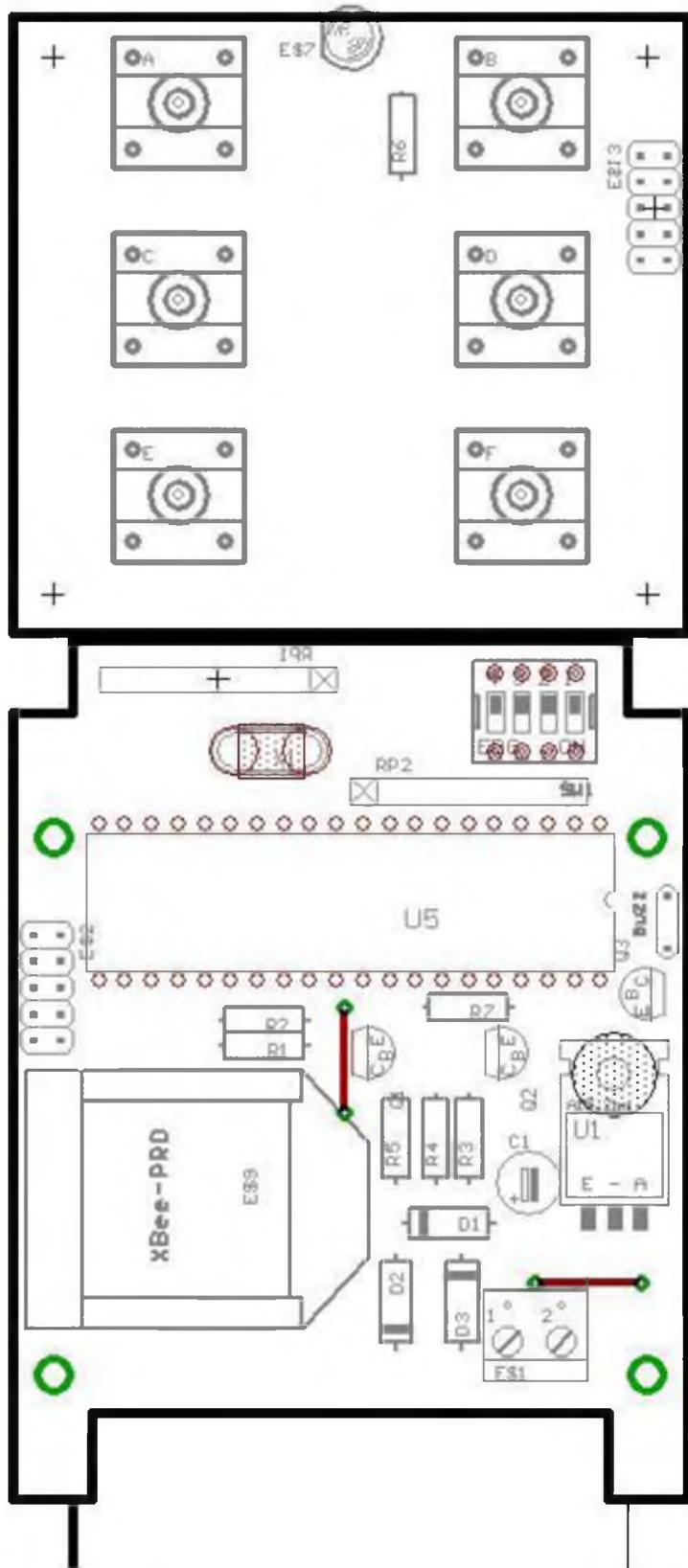


Figura 3. 17: Vista de la placa PCB del sistema de puntuación.
 Elaborado por: Gine Contreras Farah

3.3. Diseño del contador descendente.

3.3.1. Diseño del Sistema de Puntuación Electrónico.

Para el diseño y simulación de nuestro sistema contador de tiempo dedicado para la competencia de sumo, hemos utilizado la herramienta Proteus ya que cuenta con una gran cantidad de componentes actualizados en sus librerías, a continuación describiremos su funcionamiento.

Entre los componentes utilizados como podemos observar en la figura 3.18, se encuentran un microcontrolador PIC 18F4550, registros de desplazamiento 4094, display de 7 segmentos, resistencias y un pulsador.

La comunicación que realiza el microcontrolador PIC a los registros de desplazamiento es por el protocolo SPI, ya que todos estos se encuentran conectados en serie y así poder reflejar el valor o cantidad en si respectivo display.

Esta simulación muestra el funcionamiento básico y para su aplicación únicamente basto con la activación de un solo display, ya que nuestro contador para dicha competencia que se llevó a cabo en nuestra institución únicamente tiene que contar hasta cinco segundos según lo dispuesto por el reglamento general del Concurso Ecuatoriano de Robótica CER 2015.

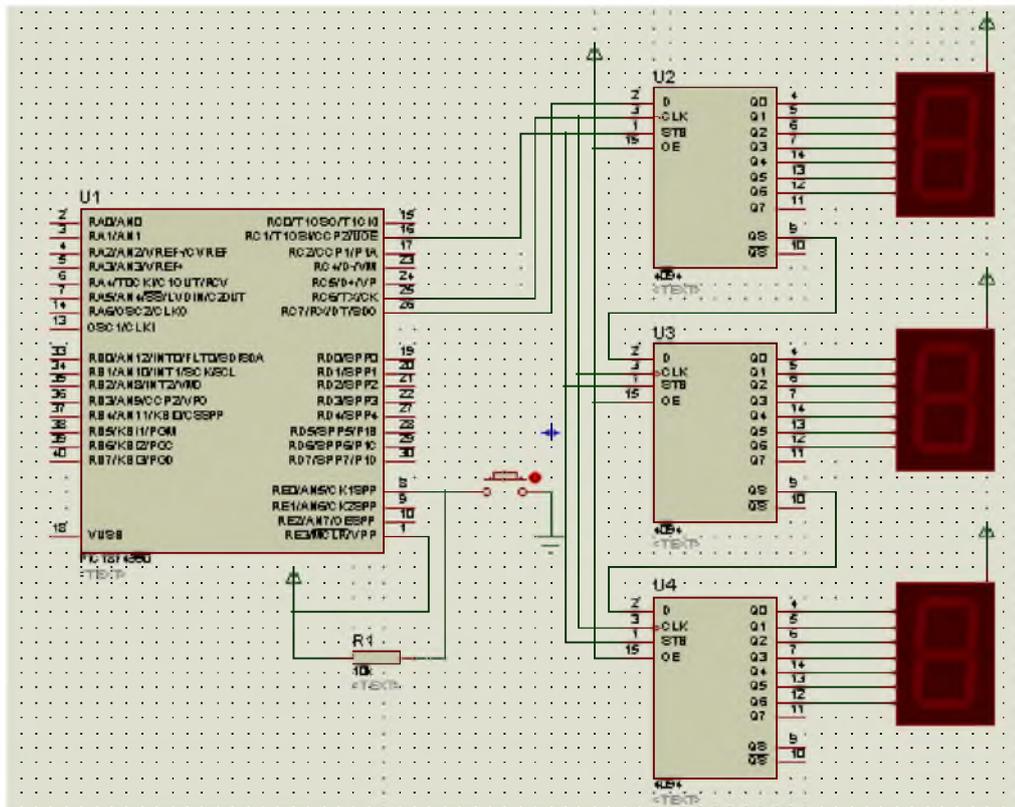


Figura 3. 18: Circuito contador de tiempo para competencia del sumo
Elaborado por: Gine Contreras Farah

Anteriormente se describió el código utilizado para nuestra aplicación, en el que describe los puertos a utilizar por nuestra comunicación SPI, así mismo los pines utilizados para el pulsador y su respectiva alimentación.

El principio de este circuito es el mismo para cualquier tipo de contador que se desee realizar, contando así unidad, decena y centena.

3.3.2. Programación de alto nivel en CCS C Compiler para el sistema de conteo ascendente.

A continuación describiremos el código que se realizó para nuestro contador de cinco segundos utilizado en la competencia de sumos.

En las primeras líneas de nuestro código como se podrá ver en la figura 3,19, están descritas como anteriormente mencionamos la llamada a la librería para el PIC con el que vamos a trabajar, también proseguimos con las configuraciones y por último el uso del protocolo SPI con sus configuraciones respectivas.

Adicionalmente están declaradas las variables globales a utilizar y declaramos las funciones que utilizaremos.

```
Sumo_tiempo_SPLc
1  #include <18f4550.h> //micropic
2  #FUSES XT
3  #use delay (clock=4000000) // o 40hz
4  #use spi(BAUD=200000, MODE=0, MSB_FIRST,DI=PIN_C2, DO=PIN_C7, CLK=PIN_C6,ENABLE=PIN_C1, BITS=8,ENABLE_ACTIVE=0)
5
6  int unidad=0;
7  int decena=0;
8  int centena=0;
9  int unidad2;
10 int decena2;
11 int centena2;
12 int z=0;
13
14 void incrementa(void);
15 int bcd_7seg(int);
16 void inicializa(void);
17
```

Figura 3. 19: Encabezado del código para el contador ascendente.
Elaborado por: Gine Contreras Farah

Continuando nuestro código como demostraremos en la siguiente figura 3,19, declaramos la función que realiza una tarea de comparación, dependiendo del número que corresponda la variable "X" se asignara por medio de una sentencia "switch" un valor hexadecimal a otra variable "Y".

```

18 int bcd_7seg(int x)
19 {
20     int y;
21     switch(x){
22         case 0:
23             y= 0x40;
24             break;
25         case 1:
26             y= 0x79;
27             break;
28         case 2:
29             y= 0x24;
30             break;
31         case 3:
32             y= 0x30;
33             break;
34         case 4:
35             y= 0x19;
36             break;
37         case 5:
38             y= 0x12;
39             break;
40         case 6:
41             y= 0x03;
42             break;
43         case 7:
44             y= 0x78;
45             break;
46         case 8:
47             y= 0x00;
48             break;
49         default:
50             y= 0x18;
51             break; }
52     return(y);
53 }
54

```

Figura 3. 20: Función de asignación de valores hexadecimal a la variable Y.
Elaborado por: Gine Contreras Farah

Para el efecto de incrementación de valores unidad y decena se requirió la creación de una función como la que se presenta en la figura 3.21, esta incrementara según sea llamada por la función principal, realizara comparaciones y dependiendo del valor que se tenga en determinado momento incrementara de unidad a decena.

```

56 void incrementa(void)
57 {
58     unidad=unidad+1;
59     if(unidad==10)
60     {
61         unidad=0;
62         decena++;
63         if (decena==10)
64         {
65             decena=0;
66             centena++;
67         }
68         if (centena==5)
69             z=1;
70     }
71 }
72

```

Figura 3. 21: Función incrementa
Elaborado por: Gine Contreras Farah

Ya que nuestro dispositivo está diseñado para un display de 7 segmentos, se tuvo que crear una función que envié los datos a la función antes mencionada “bcd_7seg” con el valor contenida en las variables unidad decena y centena, aunque en la práctica lo único que se utilizo fue la unidad ya que no es necesario la incrementación de números más allá de cinco segundos, esto lo podemos visualizar en la figura 3.22.

```

73 void inicializa(void)
74 {
75     unidad2=bcd_7seg(unidad);
76     decena2=bcd_7seg(decena);
77     centena2=bcd_7seg(centena);
78     spi_xfer(unidad2);
79     spi_xfer(decena2);
80     spi_xfer(centena2);
81 }
82

```

Figura 3. 22: Función inicializa
Elaborado por: Gine Contreras Farah

Y por último tenemos la presentación de la función principal “void main”, para el correcto funcionamiento del diseño del contador de cinco segundos.

Este código llama primeramente a la función inicializa para activar el display de 7 segmentos, a continuación la sentencia “while” entrara en un ciclo sin fin hasta que sea accionado el pin E0, para proseguir con la función de incrementación de números.

Así tendremos que llamar una a una las funciones correspondientes para luego enviar los datos resultantes por el protocolo SPI a los registros de desplazamientos 4094 que se usaron en nuestro esquemático.

```
83 void main(void)
84 {
85     inicializa();
86     while(input(PIN_E0));
87
88     while(z==0)
89     {
90         incrementa();
91         unidad2=bcd_7seg(unidad);
92         decena2=bcd_7seg(decena);
93         centena2=bcd_7seg(centena);
94         spi_xfer(unidad2);
95         spi_xfer(decena2);
96         spi_xfer(centena2);
97         delay_ms(10);
98     }
99
100 }
```

Figura 3. 23: Función principal del contador para el conteo de cinco segundos
Elaborado por: Gine Contreras Farah

CAPÍTULO 4: CONCLUSIONES Y RECOMENDACIONES.

4.1. Conclusiones.

- De acuerdo a la búsqueda de información en las competencias de robótica a nivel nacional, específicamente en las categorías de batalla de robots y sumo, el jurado llevaba las puntuaciones y el control de tiempo en forma manual. Es decir, las puntuaciones se anotaban en un papel y al final se entregaba una hoja con los resultados obtenidos en cada batalla; mientras que en la categoría de sumo el control de tiempo no era preciso para el conteo de los 5 segundos.

- A través de los diseños electrónicos utilizando el microcontrolador PIC 18F4550 se pudo lograr mediante la simulación en ISIS Proteus el control de las puntuaciones mediante seis pulsadores, los cuales fueron procesados y transmitidos vía inalámbrica por el ZIGBEE en la banda ISM de 2.4 GHz. Para el conteo de los cinco segundos se realizó el display de 7 segmentos y el circuito electrónico que permitía ejecutar la función de conteo. Durante las pruebas de las dos aplicaciones implementadas, cumpliendo con las expectativas y propósitos planteados en el trabajo de titulación.

- Los algoritmos de programación diseñados para ambas aplicaciones cumplieron con los parámetros considerados en el control y transmisión de los dispositivos implementados.

4.2. Recomendaciones.

- Incentivar a los estudiantes de esta carrera a que se interesen por los concursos de robótica para así investigar e identificar cada una de las necesidades para poder suplirlas e implementar nueva tecnología para que los concursos cada vez sean menos manuales.

- Buscar prototipos que puedan ayudar a llevar controles en áreas generales como ingreso de documentos, ingreso de puntajes, recepción de archivos o papeles, aperturas de puertas

REFERENCIAS BIBLIOGRÁFICAS

(26 de Enero de 2012). Recuperado el 04 de Enero de 2016, de Postreh:
<http://postreh.com/vmichal/thesis/figures/figures.htm>

Connie Guglielmo. (04 de Febrero de 2013). Recuperado el 10 de Enero de 2016, de Forbes: <http://www.forbes.com/sites/connieguglielmo/2013/02/04/fairchild-semiconductor-and-the-men-who-helped-invent-silicon-valley/#70241bd6618e>

Hector Hernando. (25 de Noviembre de 2010). *cuaderno de practicas g*. Recuperado el 30 de Enero de 2016, de http://cuadernodepracticassg2.blogspot.com/2010_11_01_archive.html

Kothari, C. (2004). *Research Methodology: Methods and Techniques*. New Age International.

Luis Galeano, Iván De Giacomi y Jabet Sánchez. (2008). *Entrrnadorpic16f84*. Recuperado el 30 de Enero de 2016, de <https://entrrnadorpic16f84.wordpress.com/1-arquitectura-harvard/>

Manuel Arenas. (24 de Febrero de 2014). *PC Actual*. Recuperado el 28 de Diciembre de 2015, de http://www.pactual.com/articulo/laboratorio/especiales/12206/procesadores_para_dispositivos_moviles.html

Microchip. (2006). *Microchip Technology Inc*. Recuperado el 30 de Enero de 2016, de <http://ww1.microchip.com/downloads/en/devicedoc/39632c.pdf>

Nacho. (2009). *Geocities*. Recuperado el 30 de Enero de 2016, de <http://www.geocities.ws/nachoenweb/intel.html>

Nieto, J. (15 de Diciembre de 2015). *Historia de los Computadores*. Obtenido de jinetnieto.blogspot.com

Oswaldo Barajas. (24 de Abril de 2013). *Electronicos Online*. Recuperado el 30 de Enero de 2016, de <http://www.electronicosonline.com/etiqueta/microcontrolador/>

Wylie, A. (2010). Recuperado el 20 de Enero de 2016, de Wylie: <http://www.wylie.org.uk/technology/semics/westcrel/westcrel.htm>

DECLARACIÓN Y AUTORIZACIÓN

Yo, Gine Wadih Contreras Farah, con C.C: # 1207543032 autor/a del trabajo de titulación: DISEÑO DE UN SISTEMA DE PUNTUACIÓN ELECTRÓNICO UTILIZANDO COMUNICACIÓN INALÁMBRICA ZIGBEE E IMPLEMENTACIÓN DE UN CONTADOR DESCENDENTE PARA LAS CATEGORÍA ROBOT DE BATALLA Y SUMO. previo a la obtención del título de **INGENIERO EN TELECOMUNICACIONES** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 14 de marzo de 2016

f. _____
Nombre: Gine Wadih Contreras Farah
C.C: 1207543032



REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA

FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN

TÍTULO Y SUBTÍTULO:	Diseño de un sistema de puntuación electrónico utilizando comunicación inalámbrica ZigBee e Implementación de un contador descendente para las categorías robot de batalla y sumo.		
AUTOR(ES) (apellidos/nombres):	Contreras Farah Gine Wadih		
REVISOR(ES)/TUTOR(ES) (apellidos/nombres):	M. Sc. Daniel Bayardo Bohórquez Escobar		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil		
FACULTAD:	Facultad de Educación Técnica para el Desarrollo		
CARRERA:	Ingeniería en Telecomunicaciones		
TÍTULO OBTENIDO:	Ingeniero en Telecomunicaciones		
FECHA DE PUBLICACIÓN:	14 de marzo de 2016	No. DE PÁGINAS:	76
ÁREAS TEMÁTICAS:	Electrónica, Sistemas Microcontroladores y Comunicación Inalámbricas		
PALABRAS CLAVES/ KEYWORDS:	ZIGBEE, PROTEUS, MICROCONTROLADOR, COMPILADOR, SCC C Compiler.		
RESUMEN/ABSTRACT (150-250 palabras):	<p>El presente trabajo se enfoca en el diseño de un marcador electrónico inalámbrico con comunicación ZIGBEE para utilizar como sistema de marcación de los jueces en la categoría Robot de Batalla, e implementación de un contador para controlar el tiempo de accionamiento de los robots Sumo. El uso en categoría de batalla, consiste en que el dispositivo es accionado por el jurado, mediante seis pulsadores, que son procesados a través del microcontrolador PIC 18F4550 y finalmente transmitido mediante comunicación inalámbricos ZigBee hasta la computadora, donde se visualizará el resultado obtenido durante la batalla. La ventaja de utilizar ZigBee, es que no requiere de licencia para operar en la banda libre ISM de frecuencia 2.4 GHz. Aunque ZigBee puede operar en 3 bandas de frecuencia. El propósito del trabajo de titulación, fue mejorar la forma en la que el jurado llevaba las puntuaciones de manera manual y esto dificultaba el conteo de puntos en cada robot. Mientras, que para la categoría Sumo, el control del tiempo se utilizaban celulares para ver el tiempo de 5 s en las que los robots debían atacar una vez que fue accionado el pulsador. El lenguaje de programación de alto nivel, que se utilizó fue SCC C Compiler, el software de simulación de microcontroladores es, Isis de Proteus.</p>		
ADJUNTO PDF:	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
CONTACTO CON AUTOR/ES:	Teléfono: 0986231183	E-mail: ginewcf92@hotmail.com	
CONTACTO CON LA INSTITUCIÓN:	Nombre: Palacios Meléndez Edwin Fernando		
COORDINADOR DEL PROCESO DE UTE	Teléfono: 0968366762		
	E-mail: edwin.palacios@cu.ucsg.edu.ec		

SECCIÓN PARA USO DE BIBLIOTECA

Nº. DE REGISTRO (en base a datos):	
Nº. DE CLASIFICACIÓN:	
DIRECCIÓN URL (tesis en la web):	