lProyecto 1: Microcontrolador con CPU AVR

Andres Felipe Ardila. Cód.1094954068

Diego Leonardo Cárdenas Cód.

Maycol Sabogal Ardila Cód.1115192075

Microcontroladores. Programa de Ingeniería Electrónica, Facultad de Ingeniería, Universidad del Quindío - Colombia

***Resumen – En este trabajo se implementó un microcontrolador basado en una CPU 8-bit ATMEL AVR mediante el simulador de circuitos ISIS Proteus. Se implementaron avances semanales en los cuales se empezó desde la raíz del microcontrolador para entender su funcionamiento total. Fueron construidos los puertos configurados bien sea como entradas o salidas, la memoria RAM y Flash, el timer del mismo con la correspondiente verificación del funcionamiento en cada uno de los módulos.***

***La herramienta Code Blocks fue fundamental para la verificación y prueba del microcontrolador, ya que una vez se cargaban los códigos en el ATMEGA este permitía su comprobación.***

***Palabras clave –Procesador, Temporizador, Pruebas lógicas, Memoria RAM, Puerto,direcciones,datos,proteus.***

# **INTRODUCCIÓN**

El microcontrolador fue implementado a partir de módulos de funcionamiento los cuales una vez verificados se enlazaban para así crear el componente total.

En la primera parte se construyó el diagrama de un puerto de entrada/salida según la configuración que manejen los registros. Esto también sirvió para verificar la resistencia de PULL UP, como también comprender totalmente su funcionamiento.

Seguidamente se empastillaron los puertos (8 en total) para tener un manejo más sencillo de los mismos.

El módulo siguiente comprendía la construcción de una memoria capaz de almacenar datos o instrucciones que posteriormente podrían ser utilizadas, esta es la memoria RAM. Para su correcto funcionamiento se cargaron códigos en C en el procesador ATMEGA 8515, el cual también puede verse como la memoria flash.

Para obtener un control temporal del micro, se construyó un timer encargado de esto. El timer o temporizador se basa fundamentalmente en el manejo de los tiempos, su importancia cae en la capacidad para medir un tiempo con tanta precisión como queramos (dentro de sus límites), esto se puede ver reflejado en la creación por ejemplo de señales de pulso variantes en su ancho, con respecto a la duración del mismo o como es conocida un señal PWM

# **OBJETIVOS**

## **General:**

Implementar correctamente en ISIS Proteus un microcontrolador de 8 bit con puertos bidireccionales (entrada/salida).

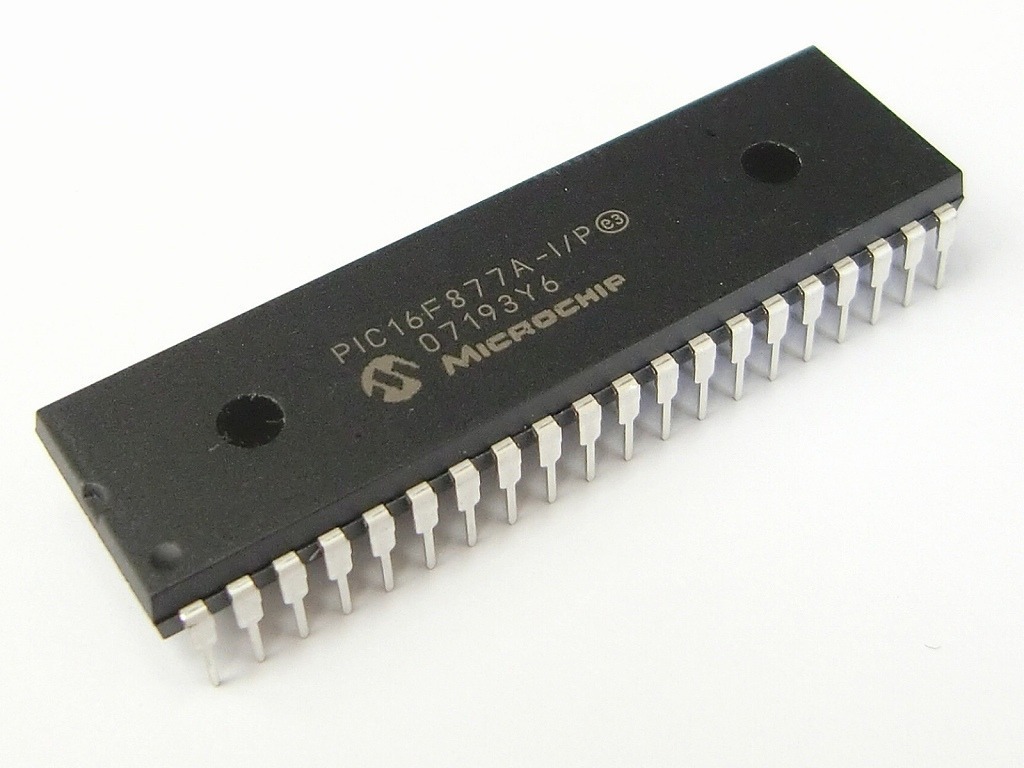
***Específicos:***

* Garantizar puertos de entrada/salida con opción de configuración.
* Memoria de programa (Flash)
* Memoria de datos
* Temporizador de 8 bits con opciones de configuración

# **MARCO TEÓRICO**

1. *Microcontrolador*

 Un microcontrolador es un circuito integrado de alta escala de integración que incorpora la mayor parte de los elementos que configuran un controlador y que contiene todos los componentes fundamentales de un ordenador, aunque de limitadas prestaciones y que se suele destinar a gobernar una sola tarea.



*Figura#1.Microprocesador*

    En su memoria sólo reside un programa que controla en funcionamiento de una tarea determinada, sus líneas de entrada/salida se conectan a los sensores y actuadores del dispositivo a controlar y, debido a su pequeño tamaño, suele ir integrado en el propio dispositivo al que gobierna.

    Un microcontrolador dispone normalmente de los siguientes componentes:

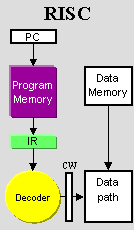
* Procesador o CPU (Unidad Central de Proceso).
* Memoria RAM para contener los datos.
* Memoria para el programa tipo ROM/EPROM/EEPROM/Flash.
* Líneas de E/S para comunicarse con el exterior.
* Diversos módulos para el control de periféricos (temporizadores, puertos serie y paralelo, CAD: Conversores Analógico/Digital, CDA: Conversores Digital/Analógico, etc.).
* Generador de impulsos de reloj que sincronizan el funcionamiento de todo el sistema.

1. *Arquitectura Risc*

La arquitectura computacional, **RISC** (reduced instruction set computer) es un tipo de microprocesador con las siguientes características fundamentales:

* Instrucciones de tamaño fijo y presentado en un reducido número de formatos.
* Sólo las instrucciones de carga y almacenamiento acceden a la memoria de datos

RISC es una filosofía de diseño de [CPU](http://es.wikipedia.org/wiki/Unidad_central_de_procesamiento) para [computadora](http://es.wikipedia.org/wiki/Computadora) que está a favor de conjuntos de instrucciones pequeñas y simples que toman menor tiempo para ejecutarse.



*Figura#2.Arquitectura RISC*

1. *Puerto entrada/salida digital*

Los puertos de microcontrolador son el punto de comunicación entre el microcontrolador y el mundo exterior, a través de ellos se pueden efectuar procesos de control electrónico sobre diferentes dispositivos, además permiten también recibir señales del mundo exterior

1. *Memoria RAM*

(RANDOM ACCESS MEMORY) - MEMORIA DE ACCESO ALEATORIO. Se utiliza para almacenar temporalmente los datos y los resultados inmediatos creados y utilizados durante el funcionamiento del microcontrolador. Es aquí donde se cargan todas las instrucciones que ejecutan el procesador y otras unidades de cómputo. Se denominan "de acceso aleatorio" porque se puede leer o escribir en una posición de memoria con un tiempo de espera igual para cualquier posición, no siendo necesario seguir un orden para acceder a la información de la manera más rápida posible.



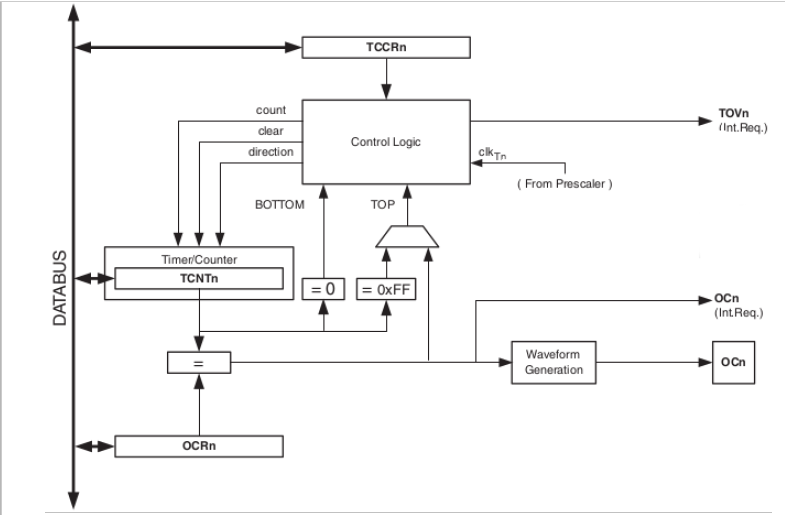
*Figura#3.Memoria RAM*

1. *Memoria flash*

Memoria programable y borrable eléctricamente, Este tipo de memoria permite reprogramar el microcontrolador sin necesidad de extraerlo del circuito, lo que se conoce comúnmente como ISP (in-system programming).

1. *Timer*

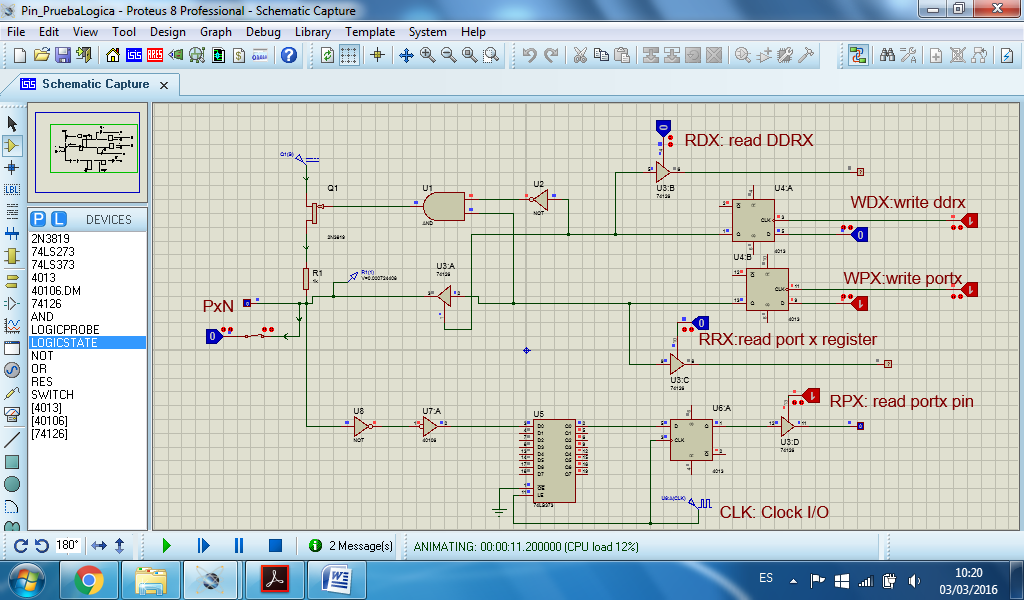
Un timer es un contador cuya entrada está conectada al reloj del sistema, Por defecto la señal que van a contabilizar los timers corresponde a la frecuencia del oscilador dividida por cuatro. Por lo tanto en realidad cuentan ciclos máquina, no ciclos de reloj. Son muy útiles para medir el tiempo que ha pasado entre dos eventos, establecer tareas para ejecutarse a intervalos regulares, etc.

**

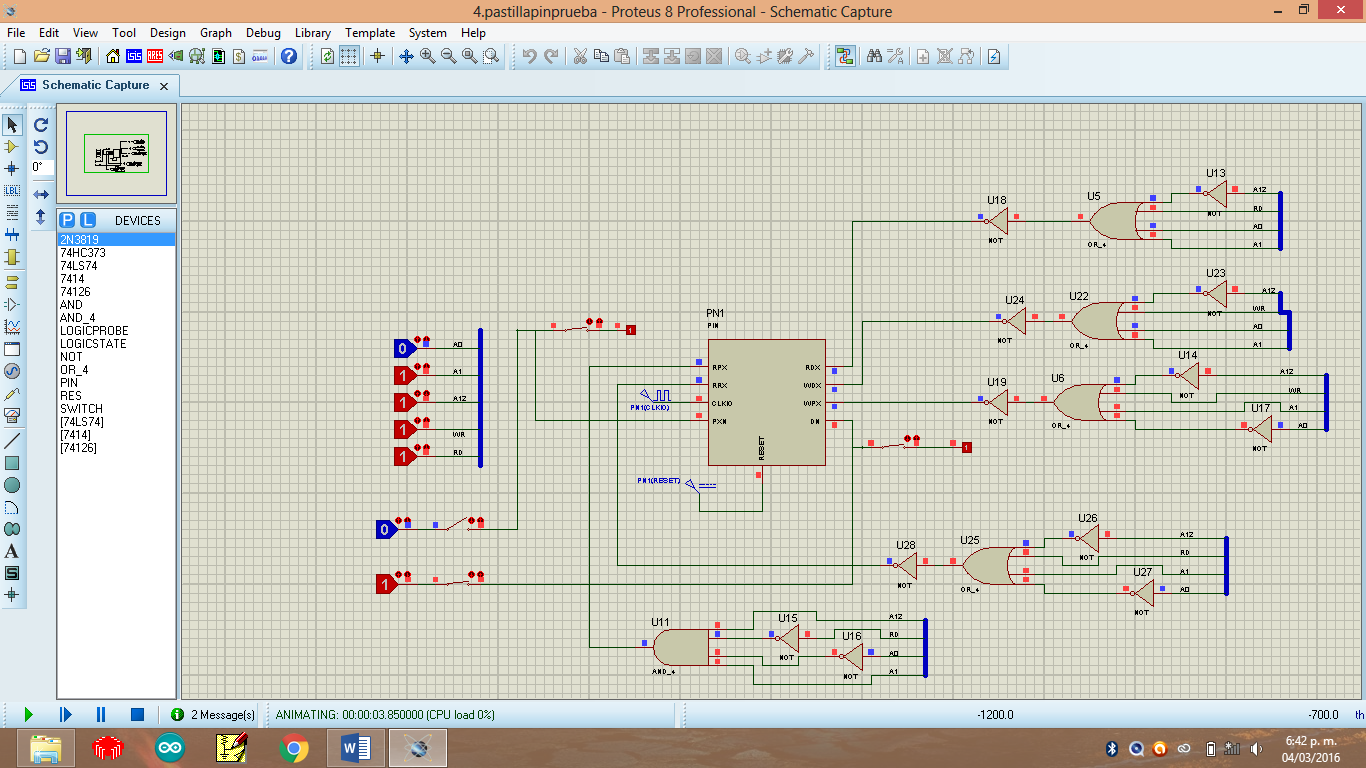
*Figura#4.Diagrama temporizador de 8 bits.*

# **PROCEDIMIENTO**

## Como primer parámetro del proyecto se realizó la construcción en el software proteus, del circuito para un solo pin correspondiente al puerto de entrada/salida, habiendo hecho el montaje, se procedió A verificar su funcionamiento mediante el uso primeramente de estados lógicos y después con la ayuda de compuertas que estaban conectados al bus de datos y generaba el valor para configurar cada uno de los registros.

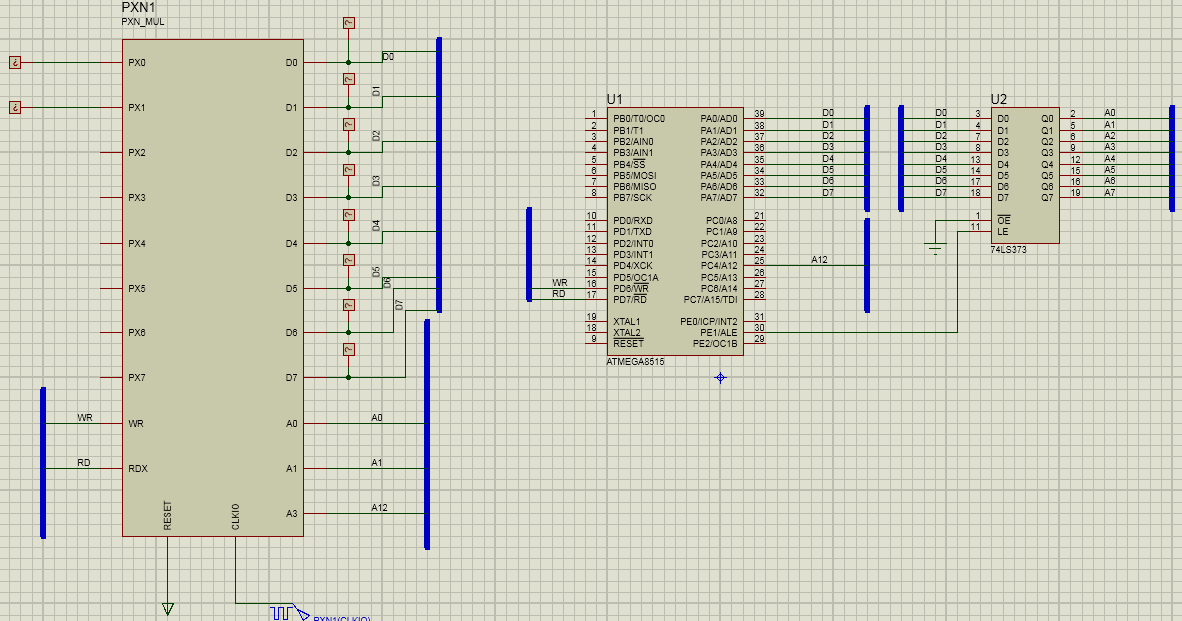


*Figura#5. Prueba pin digital con estados lógicos*



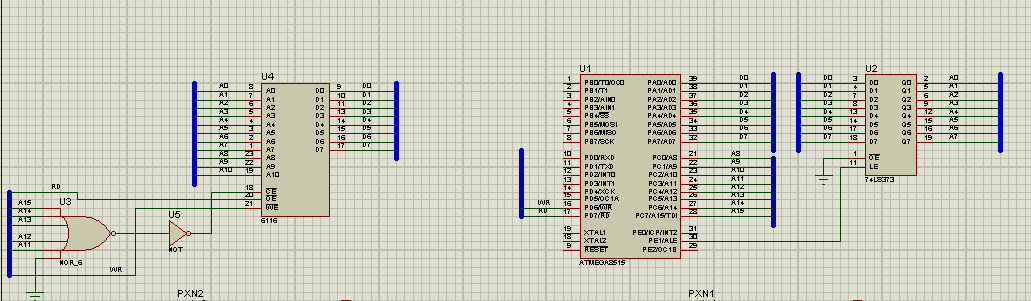
*Figura#6.Prueba circuito integrado para un solo pin*

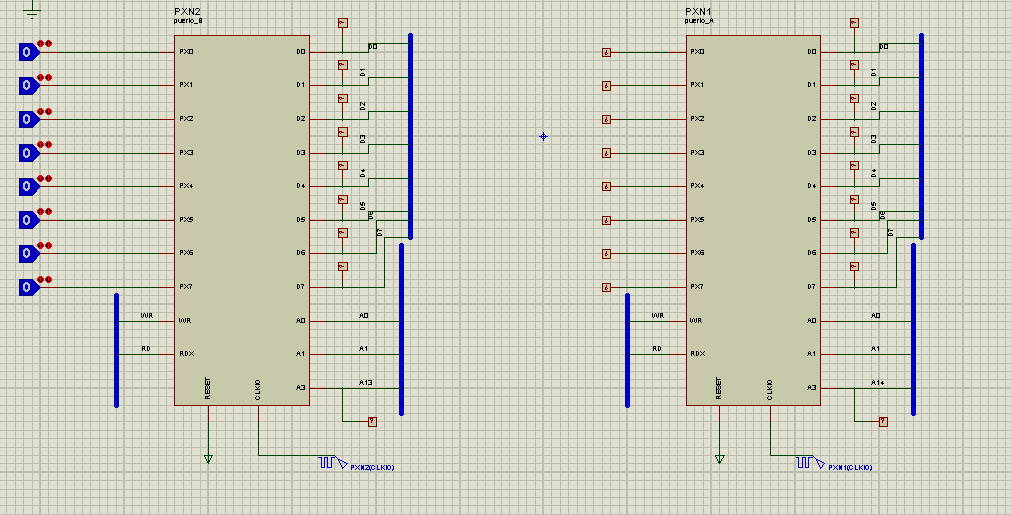
## Paso siguiente se elaboró el circuito integrado para un solo pin con su respectiva prueba, y al verificar su correcto funcionamiento se desarrolló el circuito integrado con los ocho pines correspondientes al puerto de entrada/salida del microcontrolador.

****

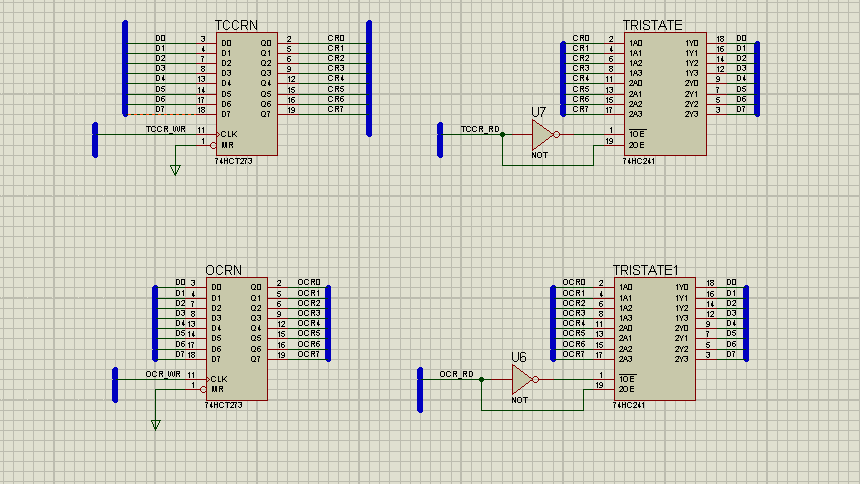
*Figura#7.Prueba circuito integrado puerto 8 pines*

## Después de tener en funcionamiento el puerto de entrada/salida para el microcontrolador, se empezaron a desarrollar las memorias RAM y flash.

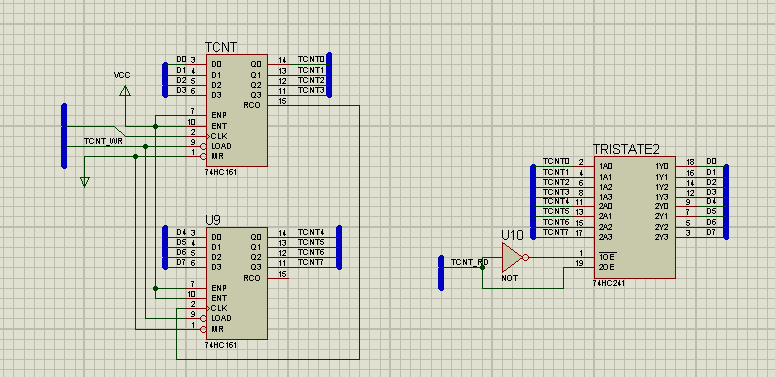




## Por último se montaría el circuito correspondiente al timer, empezando por cada uno de los registros que daban el modo de funcionamiento a esta parte del microcontrolador, como lo son el TCCRn,OCRn,TCNTn.



*Figura#8. Montaje registros TCCRn y OCCRn*



*Figura#9. Montaje registro TCNTn*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

## Como paso siguiente se implementaron los circuitos restantes del timer como el preesclador, el control lógico, el generador de forma de onda y el circuito de salida al pin y se hiso la prueba correspondiente, con los códigos propios en lenguaje c cargados al generador de datos.

## A continuación de verificar el correcto funcionamiento del timer procedimos a realizar su circuito integrado correspondiente y probarlo para estar seguros de que estaba trabajando apropiadamente.

## Como último paso del proyecto, se conectaron todos los componentes previamente generados y se verifico el correcto funcionamiento de todo el microprocesador.

# **ANÁLISIS DE RESULTADOS**

El proyecto tuvo un buen funcionamiento, en la parte de los puertos de entrada salida, los cuales se estaban configurando de forma óptima, teniendo en cuanta los datos de entrada.

De igual manera la RAM funciono de manera correcta almacenando los datos que se generaban en el puerto de entrada y escribiéndolos en el puerto de salida.

Por ultimo tenemos que decir que no pudimos saber cómo fue el funcionamiento del timer, ya que por errores de simulación, que muy probablemente se deben a las conexiones no se pudo revisar su modo de trabajo y saber si era el indicado.

# **CONCLUSIONES**

* El micro controlador según las direcciones recibidas del atmega será su forma de trabajo ya se acceder a la RAM al timer o a los puertos de entrada y salida
* En el timer podemos concluir que todo su modo de trabajo dependen de los bits del 0 al 6 del TCCRn
* En la implementación se observó como el orden de los circuitos es muy importante para poder analizar su funcionamiento también que el montaje por medio de buses de datos simplifica y da más orden al montaje
* Es muy importante tener en cuenta en qué dirección funciona cada puerta para asi poder analizar el micro antes de su simulación

# **Bibliografía**

|  |  |
| --- | --- |
|  | [1] Valdés Pérez Fernando E.,Pallás Ramón, Microcontroladores: fundamentos y aplicaciones con PIC , España:Marcombo.  [2] Hoja de especificaciones Atmega 8515. |