

Informe proyecto 1 arquitectura MCU AVR Microprocesadores.

Nathaly Sanchez Hincapie, Beymar Ruidiaz Martinez.
Universidad del Quindío, laboratorio general de electrónica, Microprocesadores

Armenia, Quindío, Colombia

ns7hincapie@gmail.com
beymar.ruidiaz.m@gmail.com

Adstract-

Resumen— Este es el informe del proyecto 1, consta de un microcontrolador con CPU AVR de 8 bits, este está constituido por una parte en el programa de simulación en el software Proteus con sus respectivos componentes como : Memoria RAM, Puertos (E/S), Latches, Timer, etc. con cada una de sus pruebas y los respectivos códigos realizados en el programa Codeblocks para poder ejecutar las simulaciones.

Palabras clave — Prescalador, microcontrolador, temporizador, Timer.

I. INTRODUCCIÓN

Partiendo de que un microcontrolador es un circuito integrado que en su interior contiene la unidad central de procesamiento (CPU), unidad de memoria (RAM), puertos de entrada y salida. Los microcontroladores están diseñados para interpretar y procesar datos en instrucciones en forma binaria. Este proyecto es la construcción de un Microcontrolador MCU AVR en el software Isis Proteus, al comenzar, se realizó el montaje de cada uno de los componentes con sus respectivos elementos, y estos, serán utilizados a través del desarrollo del proyecto, algunos de estos componentes son el puerto, los registros para las

direcciones, la Memoria RAM entre otros.

II. OBJETIVOS

- Objetivo General:
Diseñar un microcontrolador en una CPU AVR de 8 bits en la herramienta de simulación Proteus.
- Objetivo específico:
 1. Encontrar un sistema de visualización o simulación adecuado para la realización de este proyecto.
 2. Estudiar el funcionamiento de un microcontrolador elegido y sus características.
 3. Analizar los tipos y tamaños de memoria que existen.
 4. Analizar los distintos tipos de señales que ingresan o salen en los registros.

III. MARCO TEORICO

- Memoria RAM

Es un tipo de memoria de ordenador a la que se puede acceder aleatoriamente, es decir, acceder a cualquier byte de memoria sin acceder a los bytes precedentes.

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.

- Puertos (E/S)

La principal utilidad de las patas que posee la cápsula que contiene un microcontrolador es soportar las líneas de E/S que comunican al computador interno con el exterior.

Estos se destinan a proporcionar el soporte a las señales de entrada, salida y control.

- Pre-escalador:

El pre-escalador se utiliza para dividir la frecuencia de reloj y producir un reloj del temporizador.

- Timer (Temporizador):

Es un contador cuya entrada está conectada al reloj del sistema. De hecho, la mayoría de los timers pueden reconfigurarse como contadores. En ese caso, en lugar de contar pulsos de reloj cuentan los pulsos que llegan a un determinado pin. 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.

- Latches:

Es un circuito electrónico biestable síncrono que depende de la entrada de disparo denominado reloj usado para almacenar información en sistemas lógicos digitales.

- Registro DDRx:

sirve para hacer que el pin del puerto quede de lectura o no lectura, dependiendo el bit que se ingrese, si se escribe un 1 en la posición del pin, este quedará en salida, si se escribe un cero, el pin quedará de entrada.

- Registro PORTx:

Si el pin del puerto está configurado como salida, el PORTx puede hacer una escritura siempre y cuando esté configurado de tal forma mandándolo a la salida del Pxn.

- Registro PINx:

Reconoce constantemente el valor en el que esté el pin ya sea que esté configurado este como entrada o como salida, éste contiene un circuito sincronizador que permite hacer la toma correcta en los flancos en los que este configurado el registro PINx mediante este el microcontrolador puede llevar valores que se hayan introducido siempre y cuando este haya querido leer dicha acción.

- Comparadores:

Son circuitos analógicos basados en amplificadores operacionales que tienen la característica de comparar dos señales analógicas y dar como salida los niveles lógicos '0' o '1' en dependencia del resultado de la comparación. Es muy útil para detectar cambios en señales de entrada de las que solamente nos interesa conocer cuando está en un rango determinado de voltaje.

- Latches:

Es un circuito electrónico biestable síncrono que depende de la entrada de disparo denominado reloj usado para almacenar información en sistemas lógicos digitales.

IV. DESARROLLO

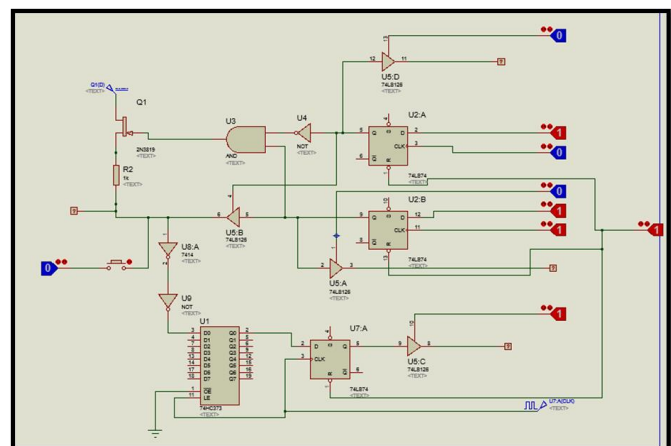


Figura 1.Circuito de un pin para un microprocesador.

Se constituyo este circuito , por los siguientes componentes :

- Flip-Flop tipo D:

Elemento de memoria que puede almacenar información en forma de un "1" o "0" lógicos.

- Logicstate:

Terminal que trabaja con variables binarias (0 y 1) permitiendo así dar la señal sea alta 1 (uno) o baja 0 (cero) para que active los componentes en el circuito.

- Compuerta lógica digital (AND):

Puerta lógica que entrega una salida alta en 1 (uno) dependiendo de los valores de la entrada, es decir, cuando la entrada (A) y la entrada (B) están en 1 (uno), su salida es 1(unos)

- Compuerta lógica digital (NOT) :

Es una puerta lógica que implementa la negación lógica. Siempre que su entrada está en 0 (cero) o en baja, su salida está en 1 (uno) o en alta, y viceversa.

- Mosfet:

Es un transistor usado para amplificar o conmutar señales electrónicas. Posee cuatro terminales llamados surtidor (S), drenador (D), compuerta (G) y sustrato (B).

- Schmitt-Triggered:

Comparador con histéresis (conserva una de sus propiedades en ausencia del estímulo que la ha generado),aplicando retroalimentación positiva a la entrada no inversora de un comparador o amplificador diferencial .

Este circuito con los anteriores componentes nombrados, hacen parte de la estructura de un pin del procesador de 8 bits, se realizo pruebas configurando los logicstate, para apreciar el buen funcionamiento de este.

Una vez realizado lo anterior y se comprobó que efectivamente el diseño del circuito y todo lo relacionado con este era correcto y estaba en óptimas condiciones, se procede a realizar el montaje de las compuertas para generar la señal, respectivamente las siguientes:

- WDX (While DDRx):

Está configurado de tal forma que los logicstate dieran como salida un (1), este flanco hace parte del registro de configuración de los DDRx.

- RDx (Read DDRx):

Hace parte de los flancos del registro de configuración.

- WPx (Portx):

Hace parte de la configuración del Pull-up.

- RRx(Read Portx)(Registro):

Hace parte de la salida o configuración del Pull-up.

- RPx (Read Portx Pin):

Hace parte de la sincronización del circuito.

- ClkIO:

Señal del reloj.

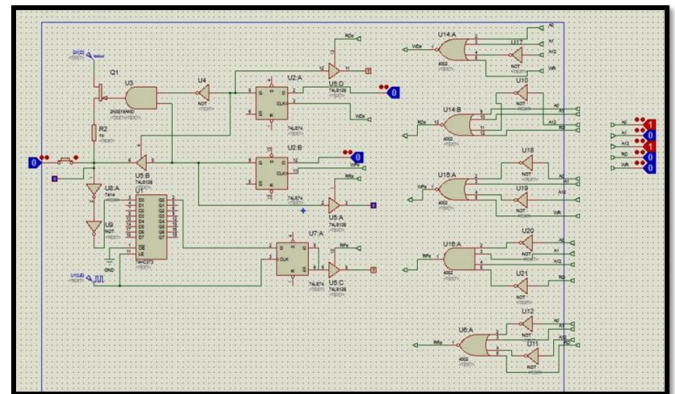


Figura2. Esquemático del circuito con las puertas lógicas AND.

Una vez realizadas estas compuertas se conectan al circuito de la Figura1, se configuran y se prueban, dependiendo del estado lógico ingresado (0 y 1) permitirá leer o escribir el dato.

Posteriormente se crean el pin con cada uno de sus pines, adicionando si es entrada, salida o bidireccional, quedando guardado dentro de los elementos de Proteus, se busca este elemento se da click en (Goto Child Sheet) y se pega el circuito de la Figura2 dentro de este-, luego de tenerlo así, se pegan ocho veces, cerciorándose de que el dispositivo funcione con las conexiones indicadas.

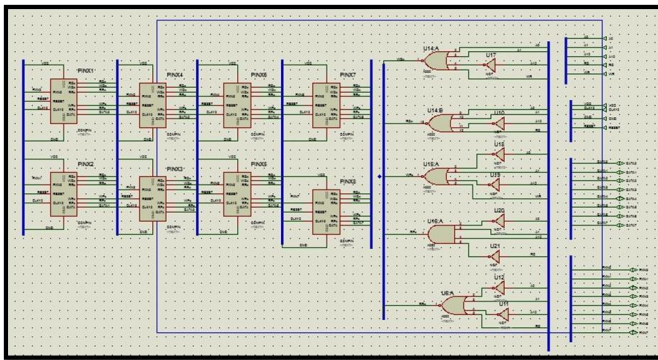


Figura3. Esquemático de los ocho pines
Luego de verificar el correcto funcionamiento del circuito, se crea el esquemático del puerto.

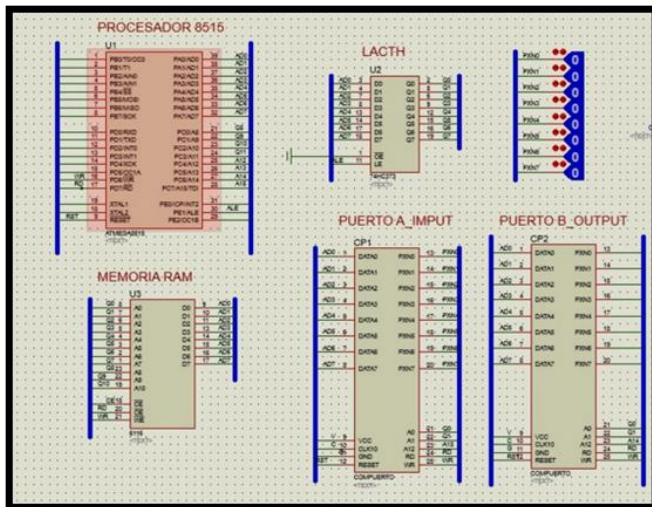


Figura4. Circuito con respectivos components.

Después de esto, se crea la Memoria RAM, se conectan los puertos (A,B), el latch, el ATMEGA 8515, para el funcionamiento correcto del circuito se anexo un código en codeblocks, para ello se carga un .HEX al procesador, este código permite corroborar el funcionamiento de la RAM .

```

4 void secuencias(uint8_t sel);
5
6 int main(){
7
8     uint8_t sel;
9
10    CPUInit();
11
12    CPUWrite(0x100, 0x01);
13    CPUWrite(0x201, 0x80);
14
15
16    CPUWrite(DDR_A, 0);
17    CPUWrite(DDR_B, 0xFF);
18
19    while(1){
20        sel = CPURead(PIN_A);
21        secuencias(sel);
22        _delay_ms(5000);
23    }
24

```

Figura5. Código de CODEBLOCK el main.

Mientras se van ejecutando estas líneas de código, el PIN_A va indica que esta leyendo y se va almacenando el sel, siendo ejecutada con un tiempo de retraso de 5000ms, este tiempo fue modificado al archivo original enviado por el profesor porque el computador no permitia ver cambios cuando se ejecutaba proteus ya que los cambios se ejecutaban muy rápido, simulando que no estuviera en acción. Incluido dentro de este código se encuentran los casos, dependiendo la entrada que se registre en Proteus, el se ejecutara.

Luego de realizar las pruebas correctas de la Memoria RAM, se realizó el montaje de los siguientes registros de control del temporizador (timer):

- Registro TCCRn:

Este pin de registro nos permite darle el funcionamiento al temporizador implementado siendo este por decirlo así, un pin de registro de control. Este registro configura la frecuencia a la que trabajara el timer.

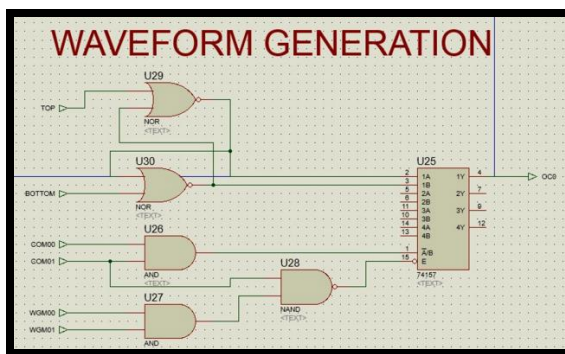


Figura11. Esquemático de generador de ondas.

- Comparadores:

Al entrar un voltaje este tiene la posibilidad de usar la referencia de voltaje interna o externa para realizar la comparación, influyendo en la señal aplicada. Basados en amplificadores operacionales que tienen la característica de comparar dos señales analógicas y dar como salida los niveles lógicos '0' o '1' en dependencia del resultado de la comparación. Es muy útil para detectar cambios en señales de entrada de las que solamente nos interesa conocer cuando está en un rango determinado de voltaje.

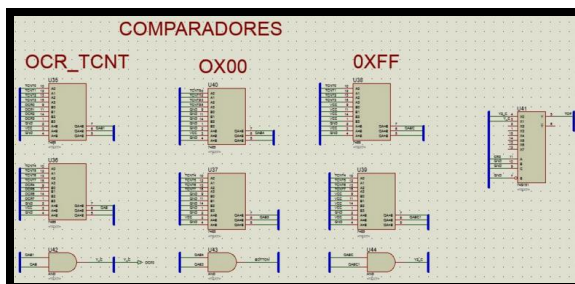


Figura12. Esquemático de comparadores.

- Pre-escalador:

El pre-escalador se utiliza para dividir la frecuencia de reloj y producir un reloj del temporizador.

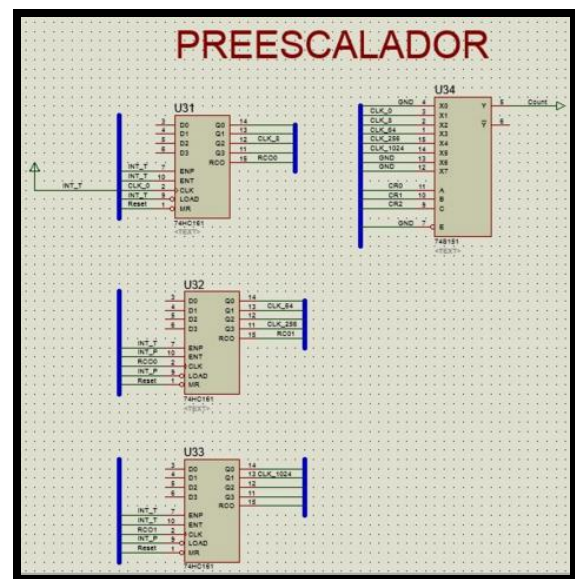


Figura13.Esquemático del preescalador.

Una vez realizado los respectivos montajes de lo anterior, se encapsulo para crear el componente Timer, respectivamente con sus patas .

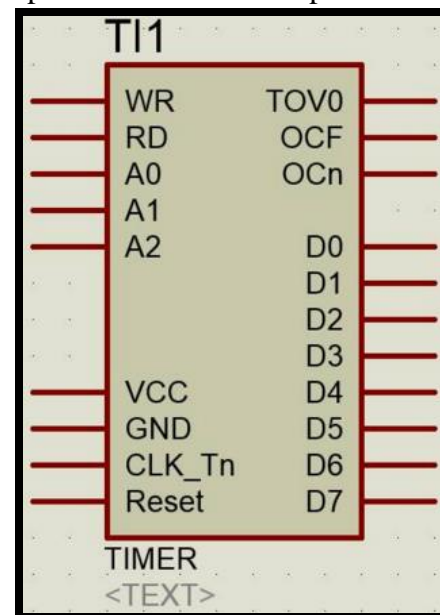


Figura14.Componente Timer.

Se realizaron las respectivas pruebas del Timer y se obtuvo una simulación sin ningún tipo de mensaje de error.

V. CONCLUSIONES

- La herramienta de desarrollo usada permitió la realización del proyecto, creando un

entorno apto entre los diseñadores y el programa.

- Permitió comprender la funcionalidad de un microcontrolador desde sus componentes internos hasta su exterior, permitiendo de esta forma hacer una idea clara del comportamiento de este sistema.
- Es importante definir los límites de conteo entre los que se puede mover el temporizador contador.

VI. REFERENCIAS BIBLIOGRAFICAS

- [1] <http://www.trescarriles.com/>
- [2] <https://es.wikipedia.org/wiki/Diodo>
- [3] <http://www.utm.mx/~hugo/Micros/Practica3.pdf>
- [4] <file:///C:/Users/NATHALY%20SANCHEZ/Downloads/Modulo%20comparador.pdf>
- [5] http://www.dte.eup.uva.es/menchu/alumnos/pfc/resumenes/Resumen%20-%20microcontroladores_8_bits.pdf



Beymar Ruidiaz Martinez. Nació en Puente nacional Santander, Colombia, el 22 de diciembre de 1986. Actualmente es estudiante de decimo semestre de Ingeniería Electrónica de la Universidad del Quindío, área de interés automatización y control, perteneció al semillero de investigación de automatización y control.

Autores



Nathaly Sanchez Hincapié. nació en Toro, Valle Colombia, en enero 6 de 1998. Se graduó de bachiller en el Colegio Corazonista. En la actualidad se encuentra estudiando ingeniería electrónica en la Universidad Del Quindío. Ponente en la VIII Jornada Técnico-científica de la Facultad de Ingeniería.