MICROCONTROLADOR DE 8 BITS.

Ciro Maya Cristian Ferney.

Sepúlveda Rojas Mario Alejandro.

**Abstract** – This report shows the procedure for the construction of an 8-bit microcontroller using Proteus as a simulation tool where digital pins were created to form two ports each with 8-pin , integrated ATMEGA8515 circuit functioning as microprocessor RAM KB and a timer for the management and control of input and output ports .

**Resumen –** El presente informe muestra el procedimiento descrito para la construcción de un microcontrolador de 8 bits utilizando proteus como herramienta de simulación, donde se crearon los pines digitales para formar dos puertos cada uno con 8 pines, circuito integrado ATMEGA8515 funcionando como microprocesador una memoria RAM de KB, y un temporizador para el manejo y el control de las señales de entrada y salida de los puertos.

**Palabras clave:** Temporizador, Registro, Contador.

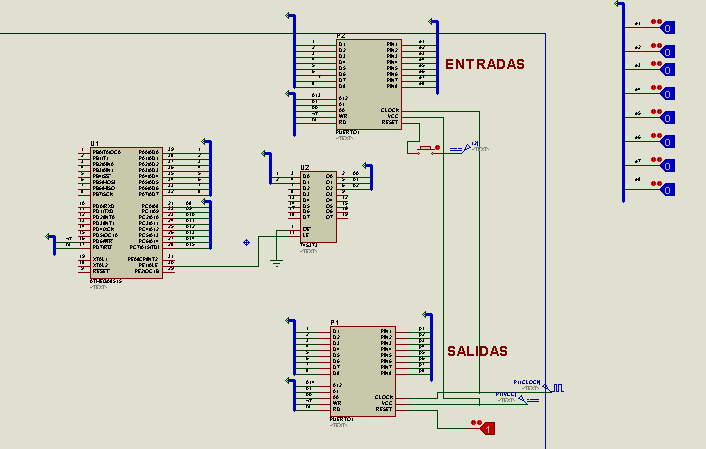
1. INTRODUCCION

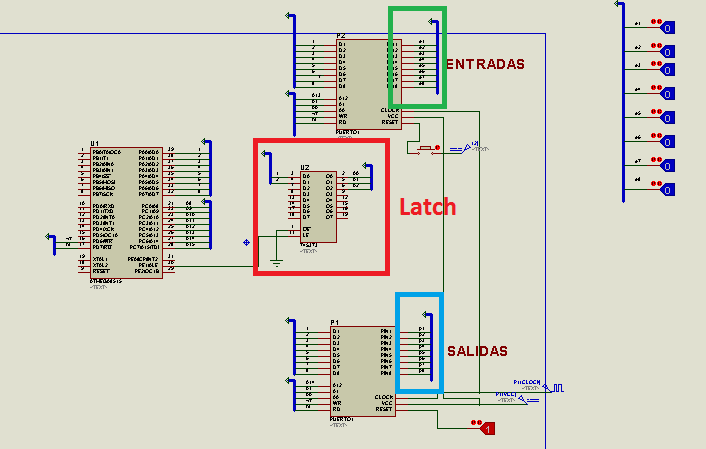
La mayoría de dispositivos electrónicos en la actualidad hacen uso de un procesador, es por eso que se hace necesario un estudio de estos elementos y una aplicación para crear un sistema basado en procesador el cual permita realizar determinadas funciones y tareas más sofisticadas a la hora de trabajar en el campo de la electrónica aplicada. Un ejemplo de ello es un microcontrolador el cual hace uso de un microprocesador internamente y otros elementos adicionales importantes como la memoria RAM, el temporizador y puertos conformados por pines digitales, que permiten la introducción y extracción de señales adentro y fuera del micro.

1. DISEÑO ANALISIS Y PROCEDIMIENTO

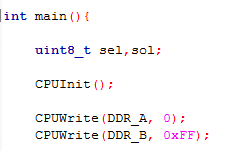
MICROCONTROLADOR AVR DE 8 bits

A continuación se mostrara el esquema del microprocesador AVR de 8 bits con el ATMEGA8515 actuando como microprocesador.





En la figura anterior tenemos configurado un puerto como entrada de datos y la otra compuerta como salida de datos.



En donde DDR\_A es la dirección 0x2000 que activa en puerto 1 de la parte superior de la imagen donde:

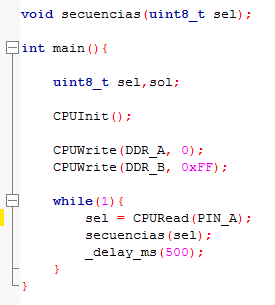
ENTRADAS: es la zona encargada de leer los datos ingresados.

El DDR\_B es la dirección 0x4000 que activa el puerto 2 de la parte inferior de la imagen, configurado como salida de datos donde:

SALIDAS: es la zona encargada de escribir los datos salientes del ATMEGA8515.

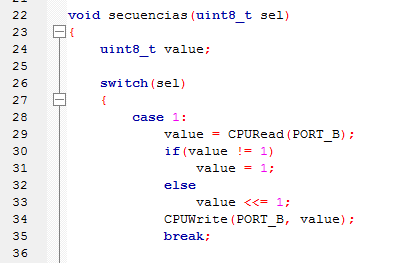
En este caso la zona demarcada con color rojo (Latch) es el componente encargado de activar la escritura o lectura de datos de cada puerto.

La programación insertada en el ATMEGA8515 es el siguiente.



En el main se configura los puertos ya antes mencionados y se lee el valor ingresado en el puerto con configuración de entrada y se almacena en la variable de tipo uint8\_t sel.

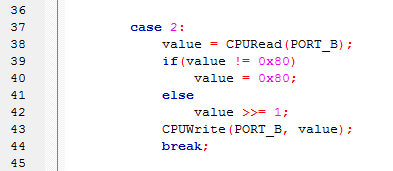
Se manda a llamar la función secuencias y se le envía como parámetro el valor de sel.



En la función se evalúa el valor de sel con un switch.

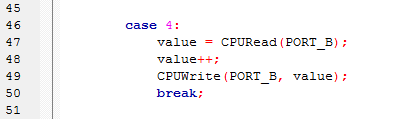
En el caso en que la entrada sea 1 (0001) se lee el valor del puerto 2, este valor se almacena en la variable de tipo uint8\_t value y se compara con el valor de sel.

Si estos valores son distintos se escribe en el puerto 2 de salida el valor 1; si son iguales se ejecuta el else en donde el valor de value es desplazado hacia la izquierda es decir, se multiplica el valor de value por dos y se escribe en la salida.

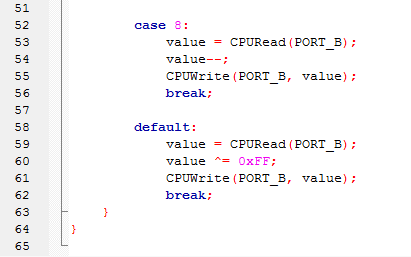


En este caso se evalúa la entrada del puerto 1, si el valor es 2 (0010), se evalúa con el valor leído en el puerto 2.

En caso de ser distinto de 0x80, que en decimal es 158(pin 7 activo), se escribe este valor en la salida. De ser igual, el valor de 158 se desplaza una posición hacia la derecha, es decir se divide por dos, que equivale a escribir un 1 en el pin 6 de salida.



Si la entrada del puerto 1 es un 4 (0100), se lee el valor del puerto 2 y respecto a este valor se empieza a sumar de 1 en 1 y se escribe en la salida.



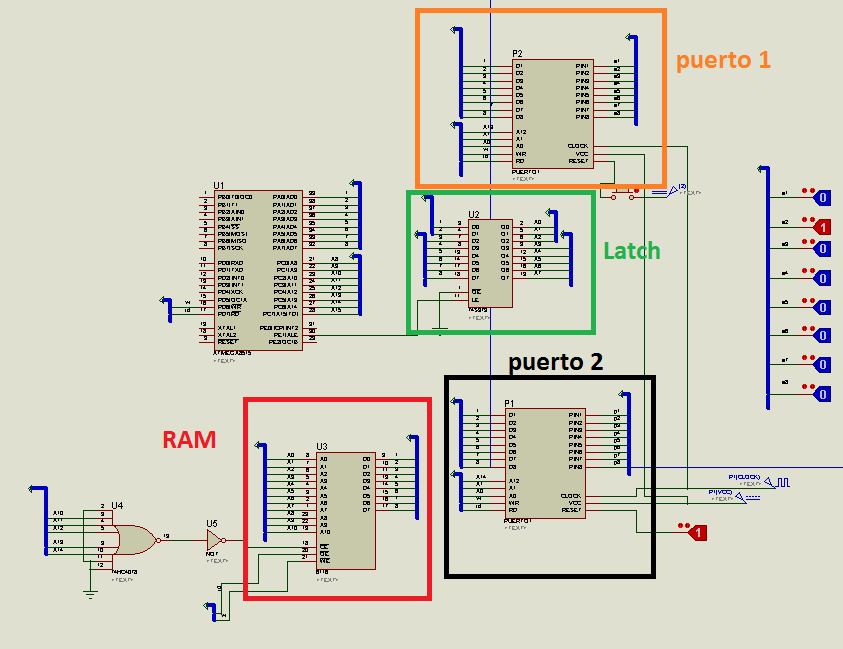
Estos dos últimos casos son el 8 y el default.

Si la entrada en el puerto 1 es 8 (1000) se lee el valor del puerto 2 y respecto a este valor se resta de 1 en 1 empezando por el valor más grande.

En caso de que la entrada no sea ninguno de estos casos se ejecuta el default que su función es negar los valores del puerto 2, es decir los que están en 1 se ponen en cero y viceversa.

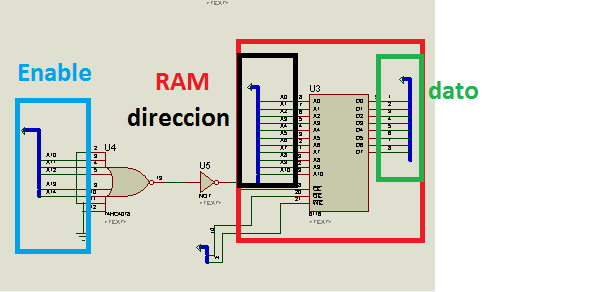
MICROCONTROLADOR AVR DE 8 BITS PRUEBA DE RAM

A continuación se mostrara el esquema del micro controlador avr de 8 bits con la RAM y su respectiva programación de prueba.



En la figura anterior tenemos:

RAM : zona demarcada de color rojo encargada de almacenar en una dirección de memoria algún dato que se le ingrese



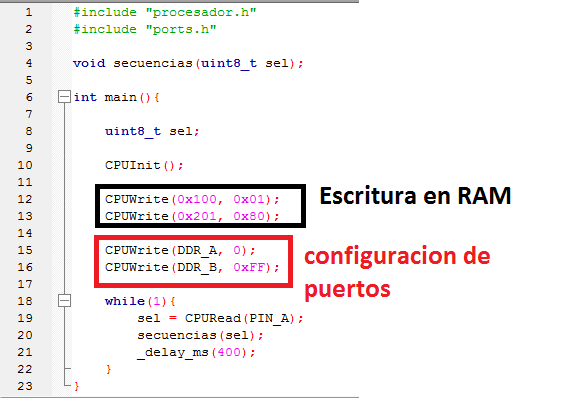
Enable : zona demarcada de color azul encargada de habilitar la RAM.

Dirección : zona demarcada de color negro. Son los pines de entrada de la dirección donde quedara almacenado el dato.

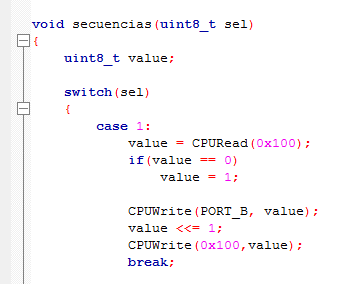
Dato : zona demarcada de color verde. Son los pines de entrada del dato a guardar.

PROGRAMACIÓN EN CODEBLOCKS

Ahora se mostrara el código que controlara la escritura y lectura de la RAM.

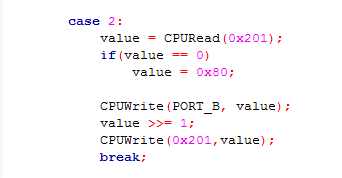


La zona demarcada de color negro, es el código encargado de escribir los valores 0x01 y 0x80 en las direcciones 0x100 y 0x201 respectivamente en la RAM. En la zona demarcada de color rojo se configura el puerto 1 como entrada de datos y el puerto 2 como salida de datos. En el ciclo while se lee el valor de entrada en el puerto 1 y se almacena en la variable sel. Luego se llama la función secuencias y se le envía como parámetro la variable sel.

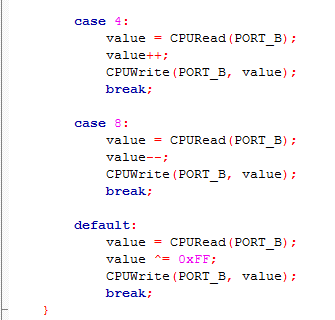


En el código anterior se evalua la variable sel.

Si este valor es 1(00000001) se lee el valor almacenado en la RAM en la dirección 0x100, si este valor es cero se escribe el valor uno en la salida, se almacena el valor en la RAM y luego es multiplicado por dos.



Si el valor es 2(00000010) se lee el valor almacenado en la RAM en la dirección 0x201, si este valor es 0x80, se escribe este valor en la salida del puerto 2, se almacena en la RAM y luego se divide entre dos.



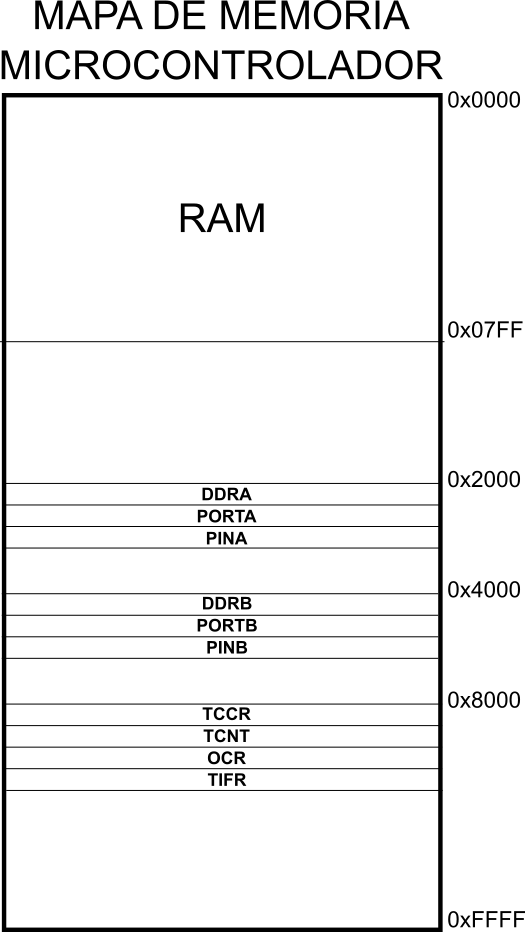
En el caso en que sea 4(00000100), se lee el valor del puerto 2 y se empieza a sumar de uno en uno a este valor leído.

Si el valor es 8(00001000), se lee el valor del puerto 2 y se empieza a restar de uno en uno al valor leído.

En caso de no ser ninguno de los valores anteriores se ejecuta el default, donde se lee el valor del puerto 2 y se niegan los valores leídos.

Temporizador 8 bits

Para la construcción del microcontrolador se tiene el siguiente mapa de memoria, puesto que se deben de tener en cuenta unos nuevos circuitos y componentes los cuales forma el temporizador de 8 bits.

 Mapa de memoria del microcontrolador.

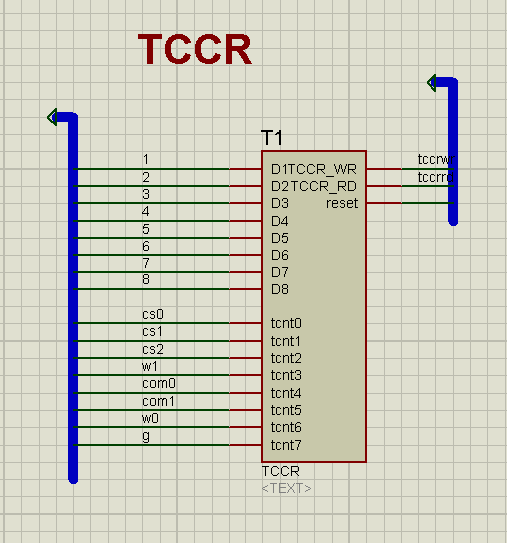
Hasta ahora se tenía conocimiento del funcionamiento del microprocesador con la memoria RAM y sus puertos. De aquí en adelante se introducirán los nuevos registros TCCR, TCNT, OCR, TIFR, los cuales tienen a su vez dos señales que vienen directamente del microprocesador para lectura y escritura (\_RD y \_WR ).

La siguiente figura muestra el diagrama del temporizador de 8 bits y las conexiones entre los diferentes registros, donde se pueden diferenciar cada uno de ellos y como es básicamente su conexión dentro del circuito.

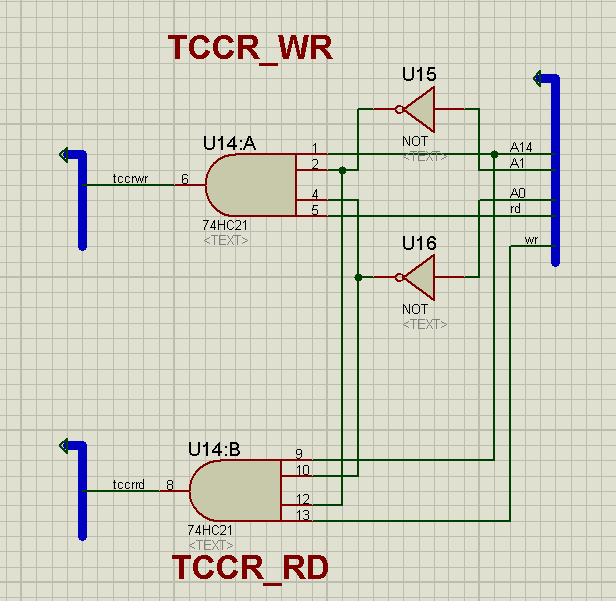
 Diagrama temporizador 8 bits.

A continuación se describirán individualmente cada registro y se profundizara en como es el circuito que los compone.

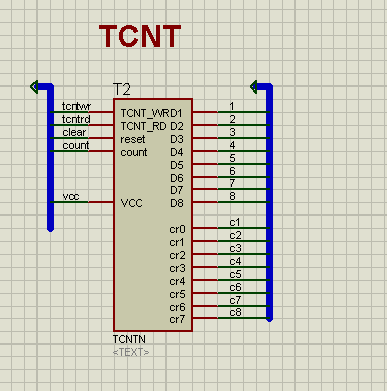
TCCR: este registro es el de control del funcionamiento del temporizador, es decir es el que determina en qué modo va a trabajar (modo normal, comparación o PWM). A este registro se le deben ingresar los datos D1-D8, las señales \_WR y \_RD del procesador para determinar si se va activar la lectura o escritura, un reset para reiniciar el registro y de este saldrán las señales que controlan los demás módulos del temporizador, como CSS, COM, WGM.



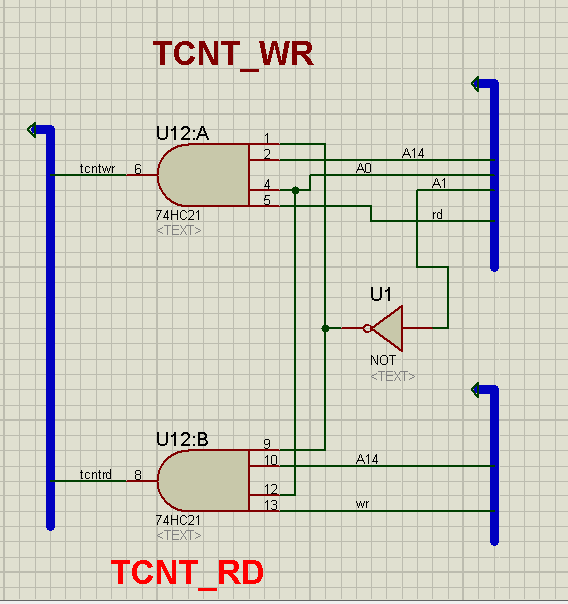
El siguiente circuito permite ver como se haría para activar las señales TCCR\_WR y \_RD que vienen conectadas al registro TCCR. Se les debe asignar una dirección desde el procesador, lo cual a e maneja en el código con el que se programe el micro.



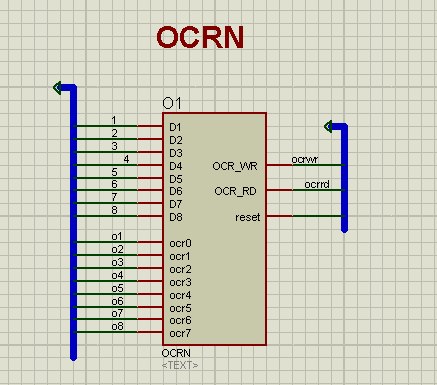
TCNT: este registro contiene la cuenta actual del temporizador, contiene los datos, las direcciones enviadas por el micro TCNT\_WR y \_RD, reset para reiniciar el registro.



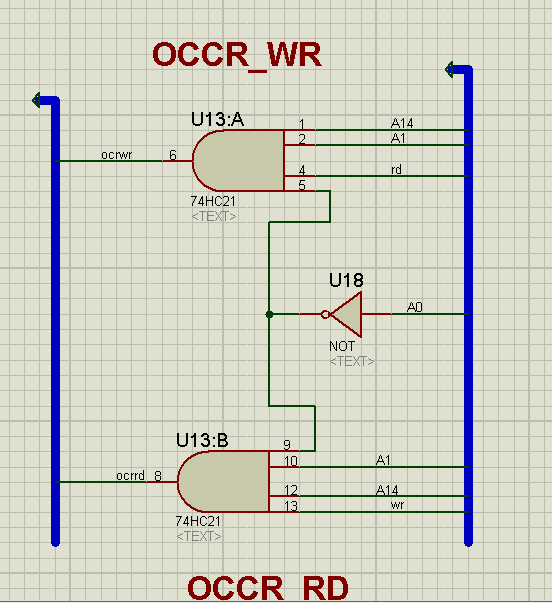
La siguiente figura contiene el circuito para activar las señales TCNT\_WR y \_RD enviadas por el procesador.



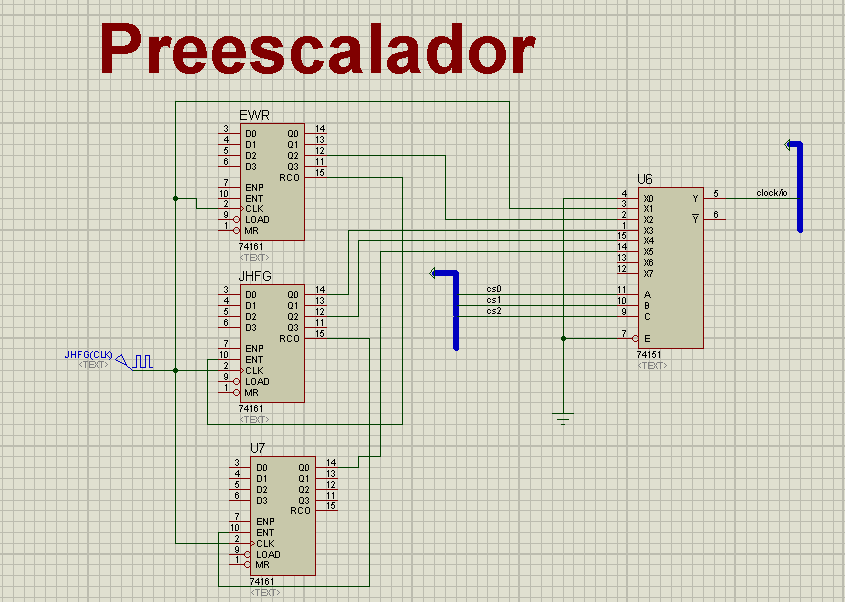
OCR: es el registro que permite almacenar un valor de comparación, el valor de este registro se compara con el valor actual del temporizador ósea con l valor que haya en TCNT, lo que permite generar interrupciones o modificar el ancho del pulso en la señal PWM.



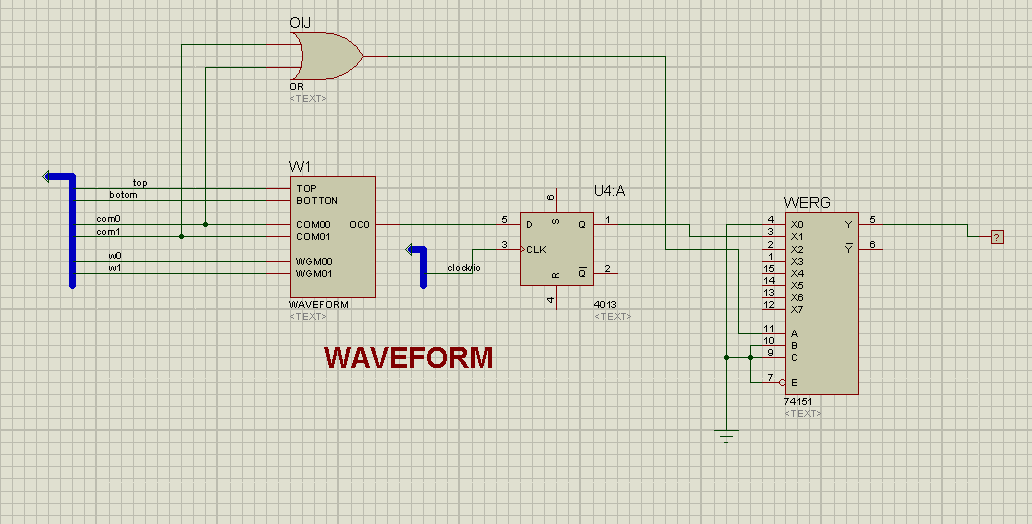
En la figura se muestra el circuito que activa las señales OCR\_WR y OCR\_RD.



PREESCALADOR: El pre escalador permite configurar la velocidad con la que se quiere contar o incrementar el contador del temporizador, para este caso se utilizó un contador de 10 bits, una señal de reloj, cuya frecuencia va a ser el máximo valor de velocidad para el contador, también se deben enviar las señales CSS0, CSS1, CSS2 provenientes del registro TCCR.



WAVEFORM: es un circuito utilizado para generar la señal de salida por el pin. Se le deben enviar las señales TOP, BOTTOM, COM0, COM1, WGM00, WGM01 provenientes del TCCR y la señal de reloj, a la salida de este circuito se encontrará la señal PWM que se desee generar.



1. CONCLUSIONES

* Se logró realizar el la simulación del microcontrolador AVR de 8bits en proteus, creando y conectando diferentes componentes como pines, puertos, memoria RAM y temporizador.
* Se pudo observar el correcto funcionamiento de cada uno de estos elementos por separado.
* No se pudo observar el funcionamiento de la salida del temporizador por errores de compilación desconocidos.

1. REFERENCIAS

[1]<https://es.wikipedia.org/wiki/Microcontrolador>

[2]<http://www.atmel.com/pt/br/products/microcontrollers/avr/default.aspx>

[3]<http://mrelbernitutoriales.com/microcontroladores-avr/>