

BEYMAR RUIDIAZ MARTINEZ

CODIGO: 1094945389

NATHALY SANCHEZ HINCAPIE

CODIGO: 98010674174

**INFORME AVANCE DE LABORATORIO DE MICROPROCESADORES**

GERARDO LOPEZ

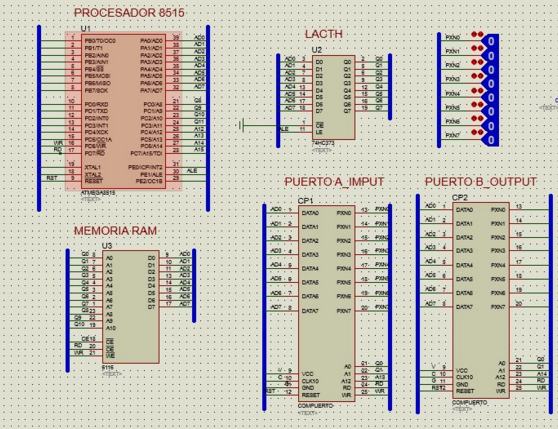
UNIVERSIDAD DEL QUINDIO

FACULTAD DE INGENIERIA

INGENIERIA ELECTRONICA

ARMENIA 2016

A continuación se explica el funcionamiento del circuito de la Memoria RAM y su anexo de código en CODEBLOCKS. Para ello es necesario cargar el .HEX al procesador, se da doble clic sobre este de la imagen de la Figura 1 (representado con el número 1 en la imagen). Posteriormente se despliega la ventana como aparece en la Figura 2. Allí se da clic en la carpeta para seleccionar el .HEX (Recuadro de color rojo), se abre una ventana como aparece en la Figura 3. Se busca la carpeta donde se haya guardado él .HEX y se carga.



6

3

5

4

2

1

Figura 1. Circuito con respectivos componentes.

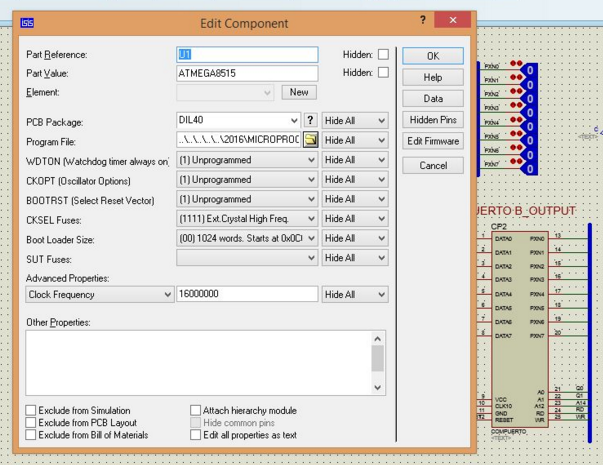


Figura 2. Opción para seleccionar la ubicación del archivo .HEX.

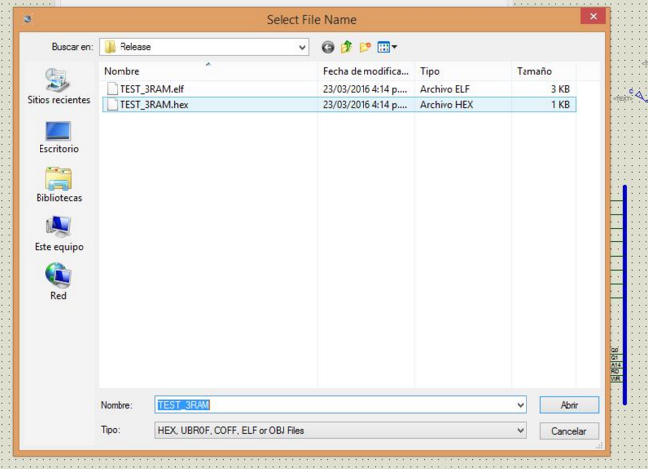


Figura 3. Ubicación del archivo .HEX

Se puede observar en la Figura 4 un fragmento del código fuente que permite corroborar el funcionamiento de la RAM; en el cuadro color rojo indica que lo que está a la derecha (0x01) va a quedar guardado en la dirección (0x100) como cuatro pines de entrada y cuatro pines de salida. En el cuadro de color morado que está formada por dos líneas de código, representa que la dirección definida como DDR\_A guarda un cero que en su forma hexadecimal representaría los ceros de entrada, es decir, los ocho pines, se ponen en entrada. Y en la dirección DDR\_B, guarda un uno, representando (0xFF) que significa que los pines, se pondrán en 1 es decir, en salida.

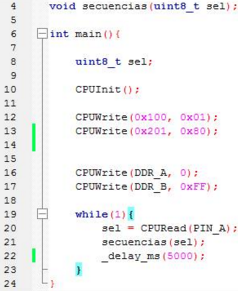


Figura 4. Código de CODEBLOCK el main.

En el cuadro verde de la Figura 4 indica que habrá un ciclo While(1), que mientras este en 1, se va a ejecutar las líneas de código que aparecen dentro, en la primer línea de código dentro del While (1) nos indica está leyendo lo que se encuentra en el PIN\_A y se almacena en sel . En la segunda línea de código en secuencia se le pasa el argumento sel que es lo que está leyendo el PIN\_A, y en la tercer línea de código el tiempo de retraso para ejecutar este ciclo es de 5000ms, este valor se modificó al código fuente enviado por el profesor ya que se descubrió que el computador no permitía ver cambios cuando se ejecutaba proteus, porque los cambios se ejecutaban muy rápido , simulando que no estuviera en acción, por lo tanto se modificaron los tiempos y se estableció en 5000ms. Todo esto va dentro de la cabeza principal que es el main.

La función void secuencias (uint8\_tsel) está conformada por un switch Figura 5 dependiendo el caso (entrada) que se registre en Proteus ,él se ejecutará. Case 1, al leer lo que se encuentra en la dirección (0x100) se guardará en value, si el valor de lo que está en value es igual a cero se pondrá todas las entradas en 1), es decir, value con su contenido 1 será escrito en el PORT\_B. Y lo que está en value será escrito en la dirección (0x100). Estas líneas de código se pueden evidenciar en el momento de ejecutar Proteus como se evidencia en la Figura 6, que a la respuesta de una entrada cero (color azul) los pines del PUERTO\_A IMPUT se ponen en uno (color rojo).

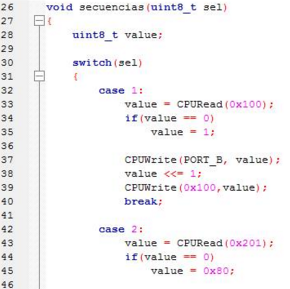


Figura 5. Código del Switch.

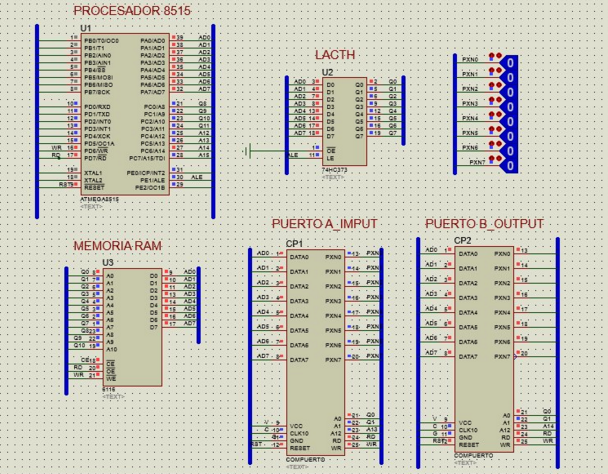


Figura 6. Proteus ejecución del código expuesto el caso 1.

En el case 4 al ingresar el valor de **1** enProteus(**00000001)** por el puerto de entrada , esto se ejecutara en la línea de código de la Figura 7 (cuadro amarillo) los valores en el puerto de salida corresponderan a la siguiente secuencia, debido a que el value estará incrementándose y este aparecerá en el PORT\_B. Como se aprecia en la Figura 8.

0000000**1**

000000**1**0

00000**1**00

0000**1**000

000**1**0000

00**1**00000

0**1**000000

**1**0000000

Su representación de lo que se ejecuta en la línea de código de la Figura 7 se evidencia en la Figura 8.

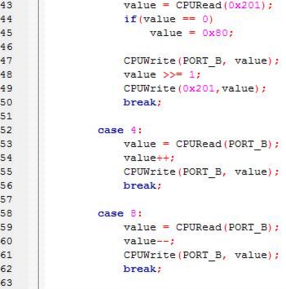


Figura 7. Código de la caso 4.

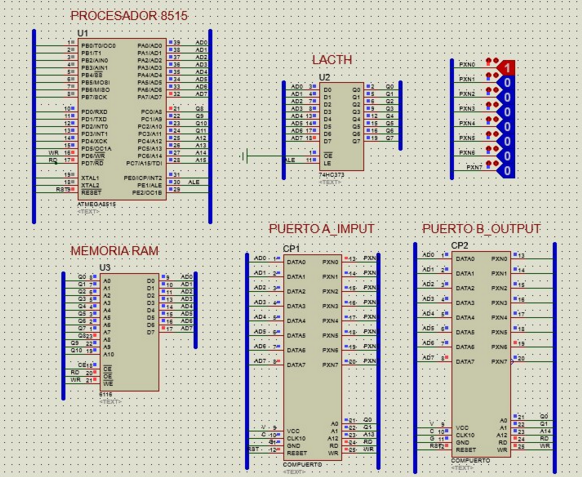


Figura 8. Proteus funcionamiento a una respuesta de una entrada en uno.

En el case 8 al ingresar el valor de **1** enProteus (**00000010)** por el puerto de entrada, esto se ejecutara en la línea de código de la Figura 9 (cuadro naranja) los valores en el puerto de salida corresponderan a la siguiente secuencia, debido a que el value estará decrementando pero su salida irá de forma ascendente y este aparecerá en el PORT\_B. Como se aprecia en la Figura 10.

**1**0000000

0**1**000000

00**1**00000

000**1**0000

0000**1**000

00000**1**00

000000**1**0

0000000**1**

Su representación de lo que se ejecuta en la línea de código de la Figura 10 se evidencia en la Figura 10.

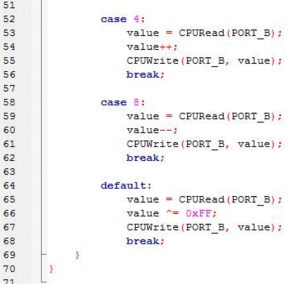


Figura 9. Código para el caso 8.

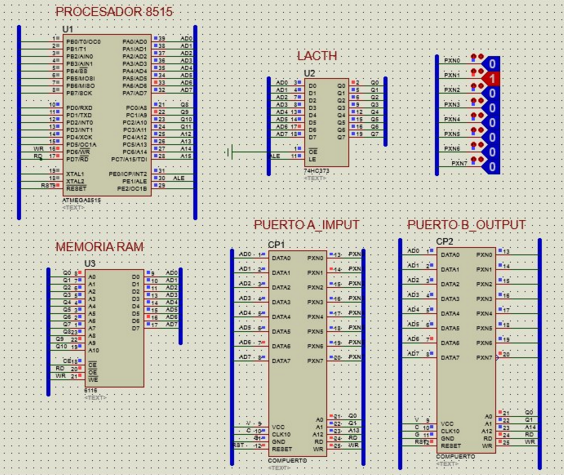


Figura 10. Configuración de proteus.