



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO



Reporte Prácticas



Integrantes de Equipo:

- Buendía Escamilla Alejandro
- Espino Ramírez Grecia Itzel
- Sergio Mendoza Parra

PROFESOR: Pérez Pérez José Juan

Grupo: 3CM7

U. de Aprendizaje: Introducción a los Microcontroladores

Semestre: 18-2

03/04/2018

ÍNDICE

1. Práctica 1	4
1.1 Marco Teórico	4
1.2 Código Fuente	6
1.3 Conclusiones.....	7
2. Práctica 2.....	9
2.1 Marco Teórico	9
2.2 Código Fuente	10
2.3 conclusiones	11
3. Práctica 3.....	12
3.1 Marco Teórico	13
3.2 Código Fuente	13
3.3 Conclusiones.....	14
4. Práctica 4.....	17
4.1 Marco Teórico	17
4.2 Código Fuente	17
4.3 Conclusiones.....	19



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO



Practica #1



Integrantes de Equipo:

- Buendía Escamilla Alejandro
- Espino Ramírez Grecia Itzel
- Sergio Mendoza Parra

PROFESOR: Pérez Pérez José Juan

Grupo: 3CM7

U. de Aprendizaje: Introducción a los Microcontroladores

Semestre: 18-2

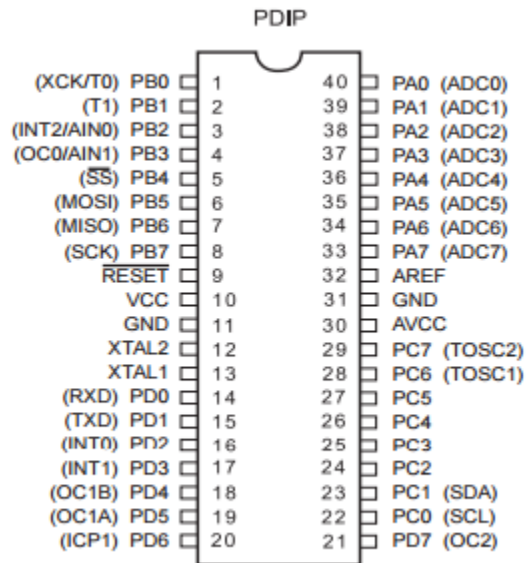
03/04/2018

1. PRÁCTICA 1

1.1 MARCO TEÓRICO

ATMEGA8535 CONFIGURACION DE LOS PINES

El microcontrolador que estaremos usando a lo largo de este curso será él; *ATMega 8535*



Micro controlador AVR de 8 bit de alto rendimiento y bajo consumo.

🌀 Arquitectura RISC avanzada.

- 130 instrucciones. La mayoría de un simple ciclo de clock de ejecución.
- 32 x 8 registros de trabajo de propósito general.
- Capacidad de procesamiento de unos 16 MIPS a 16 MHz.
- Funcionamiento estático total.
- Multiplicador On-Chip de 2 ciclos

🌀 Memorias de programa y de datos no volátiles.

- 8K bytes de FLASH autoprogramable en sistema.
Resistencia: 1.000 ciclos de escritura / borrado.
- Sección de código añadida opcional con bits de bloqueo independientes.
Programación en sistema con el programa añadido On-Chip.
Operación de lectura durante la escritura.
- 512 bytes de EEPROM.

Resistencia: 100.000 ciclos de escritura / borrado.

- 512 bytes de SRAM interna.
- Bloqueo (cerradura) programable para la seguridad del software.

Características de los periféricos.

- Dos Timer/Contadores de 8 bits con prescaler separado y modo comparación.
- Un Timer/Contador de 16 bits con prescaler separado, modo comparación y modo de captura.
- Comparador analógico On-Chip.
- Timer watchdog programable con oscilador separado On-Chip.
- Interface serie SPI maestro/esclavo.
- USART serie programable.
- Contador en tiempo real con oscilador separado.
- ADC de 10 bit y 8 canales.
 - 8 canales de terminación simple
 - 7 canales diferenciales sólo en el encapsulado TQFP.
 - 2 canales diferenciales con ganancia programable a 1x, 10x o 200x sólo en el encapsulado TQFP.
- 4 canales de PWM.
- Interface serie de dos hilos orientada a byte.

Características especiales del microcontrolador.

- Reset de Power-on y detección de Brown-out programable.
- Oscilador RC interno calibrado.
- Fuentes de interrupción externas e internas.
- 6 modos de descanso: Idle, reducción de ruido ADC, Power-save, Power-down, Standby y Standby extendido.

I/O y encapsulados

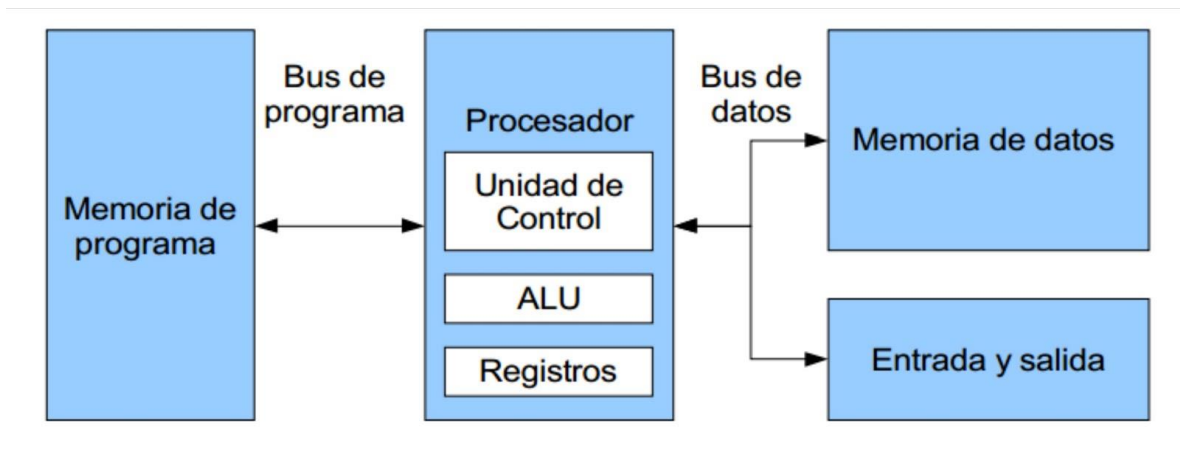
- 32 líneas de I/O programables.
- PDIP de 40 pines, TQFP y MLF de 44 pines.

Tensiones de funcionamiento.

- 2.7 - 5.5V (ATmega8535L).
- 4.5 - 5.5V (ATmega8535).

🌀 Niveles de velocidad.

- 0 - 8 MHz (ATmega8535L).
- 0 - 16 MHz (ATmega8535).



1.2 CÓDIGO FUENTE

```

.include "m8535def.inc"

.def aux=R16
.def dat=R17

ser aux
out DDRA, aux
out DDRC, aux
out PORTD, aux

aqui:
in dat, PIND
swap dat
out PORTA, dat
com dat
inc dat
out PORTC, dat
rjmp aqui
  
```

1.3 CONCLUSIONES

Espino Ramírez Grecia Itzel:

En la práctica 1 aprendí a observar el funcionamiento de algunas instrucciones mediante palabras sencillas y que por medio de la simulación se pudo ver como se cargaban cada uno de los datos en cada una de las entradas.

Así mismo se pudieron ver los registros de estado y cómo iban cambiando de acuerdo a como se ejecutaba cada una de las instrucciones paso a paso.

Buendía Escamilla Alejandro:

En esta práctica tuvimos muchos problemas ya que no sabíamos que hacia la practica de este modo tuvimos que investigar para poder saber que es lo que hacia, uan vez sabiendon lo que hacia la practica pudimos ver como se usaban los puertos de entrada y salida para obtener datos o poner datos en binario.

Mendoza Parra Sergio:

En esta practica aprendimos lo basico ya que al momento de poner el codigo binario de un numero nos mostraba su codigo en complemento a 2(la forma negative del número) y se pudo apreciar mejor con leds que al momento de poner los numeros en bianrio automaticamente se ponia en complemento a 2.



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO



Practica # 2



Integrantes de Equipo:

- Buendía Escamilla Alejandro
- Espino Ramírez Grecia Itzel
- Sergio Mendoza Parra

PROFESOR: Pérez Pérez José Juan

Grupo: 3CM7

U. de Aprendizaje: Introducción a los Microcontroladores

Semestre: 18-2

03/04/2018

2. PRÁCTICA 2

2.1 MARCO TEÓRICO

El decodificador es un dispositivo que acepta una entrada digital codificada en binario y activa una salida. Este dispositivo tiene varias salidas, y se activará aquella que establezca el código aplicado a la entrada.

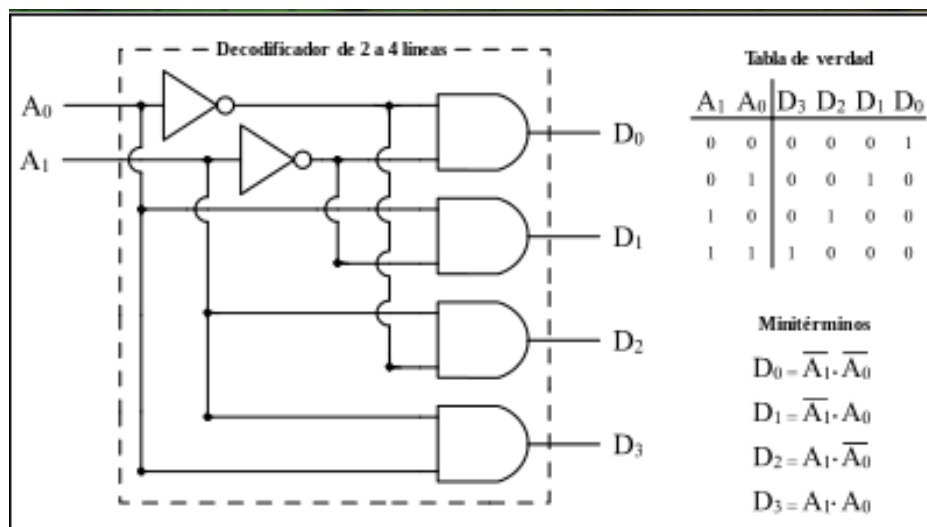
Con un código de n bits se pueden encontrar 2^n posibles combinaciones. Si se tienen 3 bits (3 entradas) serán posibles $2^3 = 8$ combinaciones. Una combinación en particular activará sólo una salida. Por ejemplo: activar la salida Q2 hay que poner en la entrada el equivalente al número 2 en binario (102).

En un decodificador de 2 a 4 (se tienen 2 pines o patitas de entrada y 4 pines o patitas de salida). En la entrada se pone el código en binario (00, 01, 10, 11), que hará que se active sólo una salida de las cuatro posibles.

Tienen como función detectar la presencia de una determinada combinación de bits en sus entradas y señalar la presencia de este código mediante un cierto nivel de salida.

Tipos de decodificadores

- Decodificadores binarios básicos.
- El decodificador de 4 bits ó decodificador 1 de 16.
- El decodificador BCD a decimal.
- Decodificadores BCD a 7 segmentos



2.2 CÓDIGO FUENTE

Esta práctica consistió en la aplicación de un decodificador, para que a través de una entrada (dada por un dip-switch) nosotros a través del uso de un programa en el micro controlador pudiéramos mostrar el número introducido en un display.

- 1) La parte 1 consistió en decodificar la entrada (números de 0-9 en binario) y mostrar su respectiva representación decimal en un display de 7 segmentos.

```
.include "m8535def.inc"          ldi R27,$27
.def aux=R16                     ldi R28,$7f
.def dato=R17                    ldi R29,$6f
ser aux
out ddrb,aux                      otro:
out ddra,aux                      ldi zl,20
                                  in dato,pina
ldi R20,$3f                       andi dato,$0f
ldi R21,$06                       add zl,dato
ldi R22,$5b                       ld dato,z
ldi R23,$4f                       out portb,dato
ldi R24,$66                       rjmp otro
ldi R25,$6d
ldi R26,$7d
```

- 2) La parte 2 consistió en la decodificación de entradas dadas por un binario del 0-F (0-15) y mostrar su respectiva representación hexadecimal en un display 7 segmentos.

```
.include "m8535def.inc"          mov R0, R16
.def aux=R16                     ldi R16,$06
.def dato=R17                    mov R1, R16
                                  ldi R16,$5b
ser aux                           mov R2, R16
out ddrb,aux                     ldi R16,$4f
out ddra,aux                     mov R3, R16
                                  ldi R16,$66
ldi R16,$3f                      mov R4, R16
```

ldi R16,\$6d	
mov R5, R16	ldi R16,\$7C
ldi R16,\$7d	mov R11, R16
mov R6, R16	ldi R16,\$39
ldi R16,\$27	mov R12, R16
mov R7, R16	ldi R16,\$5E
ldi R16,\$7f	mov R13, R16
mov R8, R16	ldi R16,\$79
ldi R16,\$6f	mov R14, R16
mov R9, R16	ldi R16,\$71
ldi R16,\$77	mov R15, R16
mov R10, R16	otro:
	ldi zl,16
	in dato,pina

- 3) La parte 3 consistió en la decodificación de una entrada en hexadecimal y mostrar su respectiva representación en ASCII.

2.3 CONCLUSIONES

Espino Ramírez Grecia Itzel:

En la práctica 2 aprendí a configurar los puertos como salidas y como entradas, además de habilitar los ocho pull-ups para cada registro y de esta forma ver cómo funcionaban con los leds. Para después hacer que solo la mitad de bits de cada registro invirtiera su valor, es decir, cambiar unos por ceros y viceversa y observar que se realizara el cambio, para que al final se obtuviera la suma de ambos valores y su resultado se mostrara.

Mendoza Parra Sergio:

En esta práctica aprendimos a como intercambiar los nibbles de esta manera una vez intercambiados los nibbles procedíamos a sumar los resultados y se mostraban en leds

Buendia Escamilla Alejandro:

Esta práctica estuvo interesante ya que pudimos apreciar y entender mas instrucciones del microcontrolador y esto para intercambiar nibbles.



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO



Practica # 3



Integrantes de Equipo:

- Buendía Escamilla Alejandro
- Espino Ramírez Grecia Itzel
- Sergio Mendoza Parra

PROFESOR: Pérez Pérez José Juan

Grupo: 3CM7

U. de Aprendizaje: Introducción a los Microcontroladores

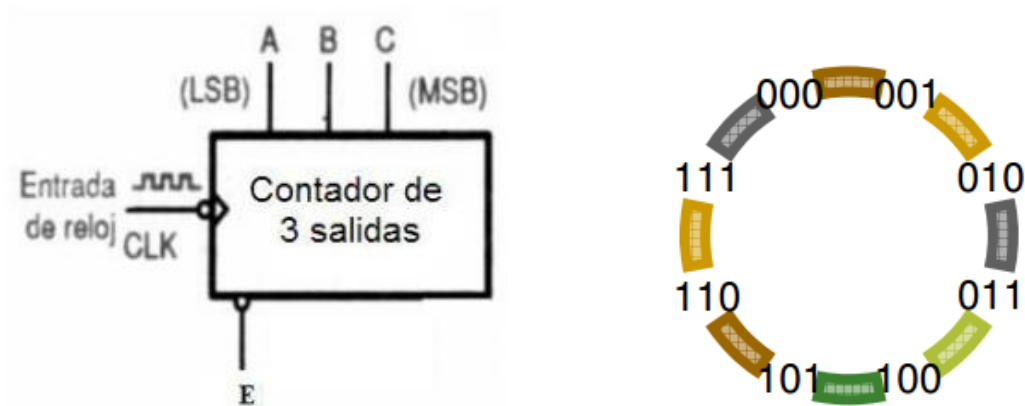
Semestre: 18-2

03/04/2018

3. PRÁCTICA 3

3.1 MARCO TEÓRICO

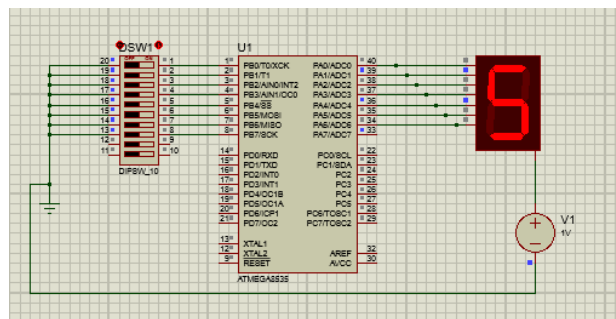
Un contador es un circuito en el que sus salidas siguen una secuencia fija que cuando acaba vuelve empezar, o circuitos que reciben sus datos en forma serial ordenado en distintos intervalos de tiempo, los pulsos de entrada pueden ser pulsos de reloj u originarse en una fuente externa y pueden ocurrir a intervalos de tiempo fijos o aleatorios. El número de salidas limita el máximo número que se puede contar.



Este circuito puede contar hasta el número 7, es decir, 8 valores (2³). Cuando llegue al último valor volverá a contar desde el principio.

A diferencia de otros circuitos como este se realizó con un microcontrolador ATMEGA8535 se tuvo que manejar los retardos que hay en el microcontrolador con LOOP los cuales debe haber varios para sincronizar la frecuencia, es decir, el tiempo con el que aparecen los valores del display. Igualmente se utilizó la correspondencia entre los valores hexadecimales y decimales para mostrarlos en el display.

3.2 CÓDIGO FUENTE



.include "m8535def.inc"	add zl,dato
.def dato=R16	ld dato,z
.def aux=R17	out PORTC,dato
ser aux	rjmp otro
out DDRC,aux	ponC: cpi dato,13
out PORTB,aux	brsh ponD
ldi R18,\$77	ldi dato,\$36
ldi R19,\$41	out PORTC,dato
ldi R20,\$6E	rjmp otro
ldi R21,\$6B	ponD: cpi dato,14
ldi R22,\$59	brsh ponE
ldi R23,\$3B	ldi dato,\$4F
ldi R24,\$3F	out PORTC,dato
ldi R25,\$71	rjmp otro
ldi R26,\$7F	ponE: cpi dato,15
ldi R27,\$79	brsh ponF
ldi R28,\$7D	ldi dato,\$3E
ldi R29,\$1F	out PORTC,dato
//ldi R26,\$36	rjmp otro
//ldi R27,\$4F	ponF: cpi dato,16
//ldi R28,\$3E	brsh Error

3.3 CONCLUSIONES

Espino Ramirez Grecia Itzel:

En la práctica 3 se realizó un direccionamiento indexado, es decir, ahora ya con ayuda de un display, cargar los valores para que se mostraran los números del cero al nueve y que de acuerdo a lo que se marcara en el display se mostrara correctamente. El direccionamiento indexado nos permite calcular la dirección efectiva por medio de algunos de los registros y la dirección efectiva se obtiene como suma de dos valores, que son el contenido del registro y un offset sin signo.

Mendoza Parra Sergio:

Para esta práctica se utilizó los loops que para un microcontrolador es como la señal de reloj, tiene el mismo funcionamiento el cual muestra en el display el valor del número a contar, sin embargo, tiene una función diferente hecha con loop donde la función más importante es calcular un buen tiempo para que se muestre de manera adecuada lo que hay en el display, ya que si no es así no se alcanza a apreciar el número en el display. Por lo cual se tomó más tiempo para la realización de la práctica para tomar el mejor tiempo.

Buendia Escamilla Alejandro:

En esta practica aprendimos a utilizar los loops ya que con ellos podemos modificar la frecuencia de trabajo para que dependiendo lo que queramos hacer podamos trabajar en esa frecuencia.



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO



Practica # 4



Integrantes de Equipo:

- Buendía Escamilla Alejandro
- Espino Ramírez Grecia Itzel
- Sergio Mendoza Parra

PROFESOR: Pérez Pérez José Juan

Grupo: 3CM7

U. de Aprendizaje: Introducción a los Microcontroladores

Semestre: 18-2

03/04/2018

4. PRÁCTICA 4

4.1 MARCO TEÓRICO

Es un elemento digital que funciona a base de estados lógicos, con los cuales indica una salida determinada basándose en un dato de entrada característico, su función operacional se basa en la introducción a sus entradas de un número en código binario correspondiente a su equivalente en decimal para mostrar en los siete pines de salida establecidos para el integrado, una serie de estados lógicos que están diseñados para conectarse a un elemento alfanumérico en el que se visualizará el número introducido en las entradas del decodificador. El elemento alfanumérico que se conecta a las siete salidas del decodificador también está diseñado para trabajar con estados lógicos, es un dispositivo elaborado con un arreglo de LED de tal manera que muestre los números decimales desde el cero hasta el nueve dependiendo del dato recibido desde el decodificador, a este elemento se le conoce con el nombre de display ó dispositivo alfanumérico de 7 segmentos.

Para este caso de la práctica se tuvo una tabla en la que se basó para realizar la misma donde de un lado se tenía la combinación en hexadecimal y el otro en decimal para poder cargar los datos de una mejor manera y así poder representarlos en el decodificador hexadecimal, cargando previamente los datos en los registros del 15 al 17.

4.2 CÓDIGO FUENTE

```

.include "m8535def.inc"
.def aux = R18
.def cuenta = R16
.def deco = R17
ldi aux, low(RAMEND)
out spl, aux
ldi aux, high(RAMEND)
out sph, aux
ser aux
out DDRC, aux
out portb, aux
ldi R20, 119
ldi R21, 0b01000001
ldi R22, $6e
ldi R23, $6b
ldi R24, $59
ldi R25, $3b
ldi R26, $3f
ldi R27, $71
ldi R28, $7f
ldi R29, $7b

uno: clr cuenta

dos: rcall decodifica
    out portc, deco
    in aux, pinb
    andi aux, 3
    inc aux

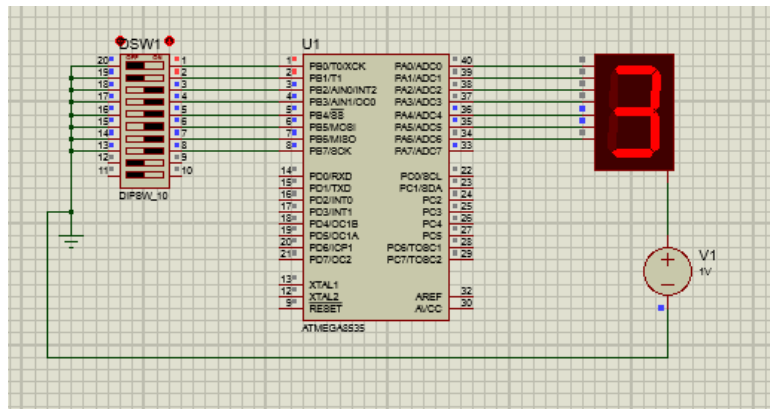
otro:
    rcall delay
    dec aux
    brne otro
    inc cuenta
    cpi cuenta, 10
    brne dos
    rjmp uno

decodifica:
    clr zh
    ldi zl, 20
    add zl, cuenta
    ld deco, z
    ret

delay:
    push aux

ldi r17, $09
WGLOOP0: ldi aux, $BC
WGLOOP1: ldi r19, $C4
WGLOOP2: dec r19
    brne WGLOOP2
    dec aux
    brne WGLOOP1
    dec r17
    brne WGLOOP0
    nop
    pop aux
    ret

```



4.3 CONCLUSIONES

Espino Ramirez Grecia Itzel:

En la práctica 4, se tuvo que hacer algo similar a la práctica 3, ya que también se necesitó de un display y de cargar los valores de los números de cero al nueve y las letras de la A a la F, es decir, los valores hexadecimales, para que después, por medio de retardos, nosotros pudiéramos modificar los valores por si queríamos que fuera más rápido o más lento en la cuenta, y que esta se repitiera sucesivamente, además de que se le agregó la funcionalidad de que si nosotros marcábamos un número o una letra fuera del rango de los valores hexadecimales, que se mostrara una E de error, indicando que ese valor no se encontraba permitido.

Mendoza Parra Sergio:

Para esta práctica se tuvo una complicación al cargar los datos a partir del número 9 para codificar los numero restantes de la A-F, ya que se habían ocupado ya una gran cantidad de registros, sin embargo, se pudo solucionar después se utilizó los códigos en la tabla donde utilizamos para guiarnos en el código hexadecimal y decimal. Esta práctica se realizó en físico y se entregó así de la misma manera, sin embargo, para fines de la realización del reporte, se realizó la misma simulada, donde se utilizó el programa de Proteus, mostrando los mismos componentes que en la práctica física.

Buendia Escamilla Alejandro:

En esta practica pudimos apreciar como con ayuda de el simulador proteus trabajaba el programa que se hizo en ensamblador para ransformar de hexadecimal a decimal esto ayudó ya que no tuvimos que conectar nada para probar si no que el simulador nos ayudó demaciado en esta practica.