

**Question 1**  
Answer saved  
Marked out of 16.00  
Flag question

Selecciona las respuestas correctas

Funciones específicas, predefinidas, Recursos limitados (memoria, potencia), La aplicación se ejecuta desde ROM o FLASH.  
La cuestión de elegir entre la implementación del hardware y el software se conoce como un tema de esta índole.

Componentes que son un denominador común en los sistemas empujados.  
en Red, de Tiempo Real, Autónomos, Móviles.

Sistemas en los que restricciones en tiempo están presentes pero no son críticos.

Modular, Escalable, Configurable, Uso de poca memoria, Soporte para CPUs, Manejadores de dispositivos.  
Confiabilidad, Costo-Eficiencia, Bajo Consumo de Energía, Eficiente uso de poder de procesamiento, Eficiente uso de Memoria, Tiempo de Ejecución apropiado.  
Sistemas computacionales con hardware estrechamente acoplado con la integración de software que están diseñados para realizar una función específica y son por lo general una parte integral de un sistema más grande.

Características de Sistemas Embebidos

Co-diseño

Procesador, Memoria Flash, Memoria RAM y Puertos E/S

Categorías de los Sistemas Embebidos

Sistemas de Tiempo Real suaves

Características deseables de los SO Embebidos

Requerimientos de los Sistemas Embebidos

Sistemas Embebidos

- 1) Características de los sistemas embebidos
- 2) Co-diseño
- 3) Procesador, ram, flash, puertos de E/S
- 4) Categorías de los sistemas embebidos
- 5) Sistemas embebidos de tiempo real-suaves
- 6) Características deseables de los SO embebidos
- 7) Requerimientos de los sistemas embebidos
- 8) Sistemas embebidos

**Question 2**  
Answer saved  
Marked out of 3.00  
Flag question

¿A qué categoría(s) de S.E. pertenece un horno de microondas?

Select one or more:

☐ Sistemas móviles

☐ Sistemas en Red

☐ Sistemas de Tiempo Real

☐ Sistemas de Autónomos

### Sistema de tiempo real, autónomo

**Question 3**  
Answer saved  
Marked out of 3.00  
Flag question

¿A qué categorías de S.E. pertenece un teléfono celular?

Select one or more:

☐ Sistemas de Tiempo Real

☐ Sistemas Móviles

☐ Sistemas Autónomos

☐ Sistemas en Red

Sistemas de tiempo real, móviles, autónomos, sistemas en red

¿Qué es lo que hace que algunos sistemas embebidos cuenten o no con sistemas operativos?

Dependerá de las tareas que deba realizar, si solo tiene que realizar una tarea en específico entonces no necesita un sistema operativo, pero si hará multitareas entonces es posible que sea conveniente que cuente con un SO que administre los recursos.

Question 4

Answer saved

Marked out of 3.00

Flag question

¿A qué categoría(s) de S.E. pertenece un dispositivo GPS?

Select one or more:

- ☐ Sistemas Autónomos
- ☐ Sistemas de Tiempo Real
- ☐ Sistemas Móviles
- ☐ Sistemas en Red

Autónomo, tiempo real, móvil.

Question 5

Answer saved

Marked out of 8.00

Flag question

¿Dónde pueden ser encontrados los sistemas embebidos?

A ▼

B

I

A ▼

Podemos encontrarlos en cualquier lugar donde se necesite automatizar una tarea específica mediante algún dispositivo programable basado en microprocesador. Por ejemplo, en autos, microondas, celulares, lavadoras, etc. Casi en cualquier lugar donde algún aparato electrónico haga alguna función programable.

Los podemos encontrar casi en cualquier lugar donde se necesite automatizar alguna tarea mediante algún dispositivo programable basado en microprocesador. Por ejemplo en autos, microondas, celulares, lavadoras, etc. Donde se pueda reemplazar algún proceso mecánico que hacía tal vez un circuito específico muy complejo, y este pueda reemplazarse por un microprocesador más pequeño, programable y flexible, y por supuesto más económico a la hora de actualizar el sistema.

Question 6

Answer saved

Marked out of 8.00

Flag question

Defina lo que es un Sistema de Tiempo Real.

A ▼

B

I

A ▼

Es aquel sistema que da una respuesta al momento, esto no quiere decir que la respuesta sea totalmente instantánea o en vivo, si no que puede haber un desfase de tiempo muy corto el cual no afecte a la percepción del usuario, lo que hace que piense que es una respuesta instantánea. La respuesta debe ser correcta para considerar a un sistema de tiempo real, además de que el tiempo debe ser aceptable de acuerdo a la tarea.

Para que un sistema sea considerado de tiempo real, la respuesta debe ser correcta, en los sistemas duros, se debe cumplir forzosamente con las restricciones de tiempo, ya que en ocasiones vidas dependen de él. En los sistemas suaves el tiempo no es tan importante pero si debe ajustarse a la percepción del usuario. El tiempo de respuesta máximo o mínimo dependerá del tipo de aplicación.

**Uso**

El sistema ABS permite mantener durante la frenada el coeficiente de rozamiento estático, ya que evita que se produzca deslizamiento sobre la calzada. Teniendo en cuenta que el coeficiente de rozamiento estático es mayor que el coeficiente de rozamiento dinámico, la distancia de frenado siempre se reduce con un sistema ABS.

Si bien el sistema ABS es útil en casi todas las situaciones, resulta indispensable en superficies deslizantes, como son pavimentos mojados o con hielo, ya que en estos casos la diferencia entre el coeficiente de rozamiento estático y el dinámico es especialmente alto.

Cuando se conduce sobre nieve o gravilla y se frena sin sistema ABS, se produce el hundimiento de las ruedas en el terreno, lo que produce una detención del coche más eficaz. El sistema ABS, al evitar que se produzca deslizamiento sobre el suelo también evita que se hundan las ruedas, por lo que en estos tipos de superficie, y deseando una distancia de frenado lo más corta posible sería deseable poder desactivar la acción del ABS.

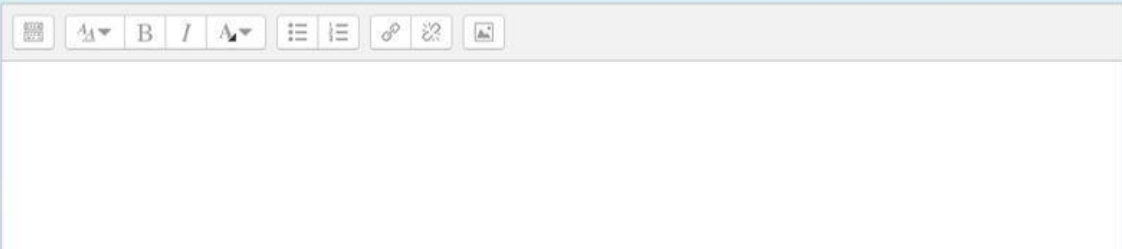
Algunos sistemas usados en autos deportivos o de desempeño, permiten al sistema del vehículo desactivar el uso del ABS para producir una frenada más brusca al principio y permitir el control del mismo con una velocidad más baja. Es decir el sistema antibloqueo entra a trabajar con retraso, permitiendo derrapes controlados o enterramientos en terrenos blandos

---

**Describe la organización de un sistemas empotrado para frenos antibloqueo**

Incluir:

- Lista de componentes, la descripción y función de cada uno (específica y simplificada)
- Categoría (argumente el porqué de la(s) categoría(s)).



Partes, a que categoría pertenece, un caso de uso del Tamagochi

Cuenta con una pantalla en la cual se muestra a la mascota y de esta manera interactúa con el usuario. Bocinas para transmitir los sonidos al mundo real, decodificador de audio mp3, memoria para almacenar los datos del programa que se está ejecutando, también para guardar los cambios, un Microprocesador que es el encargado de ejecutar las instrucciones de código para que el dispositivo cumpla con su objetivo. Botones para que el usuario ingrese datos que serán interpretados por el procesador, o para ingresar datos que se le pidan al usuario. Baterías para funcionar, ya que es un sistema embebido móvil. Antena para poder recibir la señal de otros tamagochis.

Alimentar a la mascota

El procesador extrae la instrucción de la memoria flash

El procesador decodifica la instrucción y se da cuenta que es alimentar a la mascota

El procesador manda una senial por un puerto de salida donde esta conectada la pantalla lcd

Se carga la senial a los puestos de la panatalla, y se da la instrucción de mostrar el programa espera a que el usuario reaccione y oprima una tecla, la cual será ingresada mediante un botton

el procesador leera el dato del puerto de E/s

y decodificara la instrucción a ejecutar

la instrucción es aceptar comida

el procesador ejecuta la instrucción y cambia las banderas del programa para indicar que la mascota ha sido alimentada

Móvil, autónomo, tiempo real, de red.

Tiene,

Partes, a que categoría pertenece, un caso de uso del robot de star wars

Partes, a que categoría pertenece, un caso de uso de alexa

Question 8

Answer saved

Marked out of 34.00

Flag question

a) El propósito de la ALU es:

- ☐ Interpretar los códigos de operando ☐ Controlar el bus de direcciones ☐ Calcular el número de ciclos máquina requeridos ☐ Realizar operaciones aritméticas y lógicas

b) Ocorre un conflicto en el ducto cuando:

- ☐ Un puerto de salida es habilitado por una señal de lectura ☐ Más de un dispositivo está leyendo del ducto de datos ☐ Familias CMOS y TTL están siendo utilizados en el ducto de datos ☐ Más de un dispositivo está escribiendo al ducto de datos

c) También Ocorre un conflicto en el ducto cuando:

- ☐ Familias CMOS y TTL están siendo utilizados en el ducto de datos ☐ Un dispositivo de entrada es habilitado por una señal de escritura ☐ Más de un dispositivo se les está escribiendo un dato por el procesador ☐ Más de un dispositivo está leyendo del ducto de datos

d) El ducto de datos es:

- ☐ Unidireccional y de tres estados ☐ Bidireccional y de tres estados ☐ Unidireccional y Bidireccional ☐ Ninguna de las respuestas

e) El microprocesador tipo RISC sabe cuales bytes interpretar como códigos de operandos porque:

- ☐ Cada código de operación implica el número de bytes de información que le sigue ☐ Cada dirección contiene una instrucción ☐ El programador debe especificar cuáles bytes son código de operando ☐ Cada 3 bytes existe un código de operando

f) El microprocesador tipo CISC sabe cuales bytes interpretar como códigos de operando porque:

- ☐ Cada byte es un código de operando ☐ El programador debe especificar cuales bytes son código de operando ☐ Cada código de operación implica el número de bytes de información que le sigue ☐ Cada 3 bytes existe un código de operando

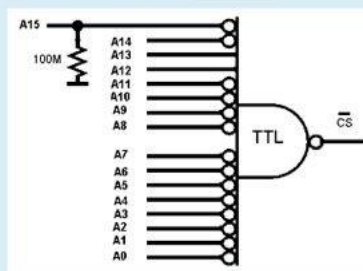
g) Los sistemas basados en microprocesadores se comunican con los periféricos mediante:

- ☐ Las memorias de solo lectura ☐ Un teclado ☐ El ducto de direcciones ☐ Los puertos de entrada y salida

h) En un sistema basado en microprocesador que utiliza dispositivos de memoria de 64K bytes, ¿Qué líneas de direcciones se conectan al IC de memoria?:

- ☐ A0-A9 ☐ A10-A15 ☐ A0-A15 ☐ A0-A7

i) Determina la dirección(es) a la(s) que responde el siguiente decodificador:  H



j) Los dispositivos de tres estados se utilizan porque:

- ☐ Estos permiten que varios dispositivos puedan conectarse juntos fácilmente ☐ Consumen poca potencia ☐ Estos proveen un tercer estado lógico válido ☐ No son costosos

k) Los dispositivos E/S en un sistema basado en microprocesador:

- ☐ Todas las respuestas ☐ Responden a una dirección ☐ Tienen direcciones específicas asignadas a ellos ☐ Son controlados por un circuito decodificador de direcciones

l) Los sistemas basados en uP y uC son ms flexibles que los diseños lógicos debido a:

- ☐ Utilizan dispositivos LSI ☐ Poseen mayor velocidad ☐ El hardware es especializado ☐ Su operación es controlada por software

m) Las memorias RAM no se utilizan para almacenar datos por tiempo prolongado porque:

- ☐ El contenido no puede ser modificado ☐ Su contenido se pierde cuando se dejan de energizar ☐ Son muy lentas ☐ Son muy costosas

n) Las memorias FLASH y ROM se utilizan principalmente para:

- ☐ Almacenar permanentemente programas y datos ☐ Almacenar datos ☐ Realizar lecturas y escrituras ☐ Almacenar datos y programas temporalmente

o) Los periféricos son:

- ☐ El microprocesador ☐ Los programas ☐ Los dispositivos de Memoria ☐ Los dispositivos de entrada y salida

p) Los sistemas basados en uP o uC manejan datos en grupos de 4 bits llamado:

- ☐ Byte ☐ Half Word ☒ Nibble ☐ Word

q) En un sistema basado en microprocesador con un ducto de direcciones de 20bits, ¿Cuál es el número máximo de dispositivos de memoria de 1K byte que puede contener?:



Question 9

Answer saved

Marked out of 5.00

Flag question

Seleccione el lenguaje acorde a la descripción:

Lenguaje que será convertido

Lenguaje que solamente es una representación simbólica de un lenguaje numérico

Lenguaje numérico baso en 1's y 0's entensible y ejecutable por uP y uC

Lenguaje al que se convertirá

Lenguaje que utiliza una notación especial orientada al problema a resolver y no se requiere conocimiento del código máquina

Lenguaje Objeto

Lenguaje Ensamblador

Lenguaje Objeto

Lenguaje Máquina

Lenguaje de alto nivel

Question 10

Answer saved

Marked out of 5.00

Flag question

Selecciona la opción acorde a la descripción:

Grupo de instrucciones que desarrollan una tarea, puede usarse dentro del programa varias veces pero se almacena solamente una vez en memoria, y generalmente regresa valor(es).

Permite asignar un nombre a una porción de texto y reutilizarla en el programa las veces que sea necesario.

Grupo de instrucciones que desarrollan una tarea, puede usarse dentro del programa varias veces pero se almacena solamente una vez en memoria.

procedimiento

macro

función

**Question 1**

Not yet answered  
Marked out of 3.00  
Flag question

Selecciona la respuesta correcta:

PIN	SIGNAL	MODO MINIMO	MODO MAXIMO
1	GND	Vcc	
2	A14	A15	
3	A13	A16/S3	
4	A12	A17/S4	
5	A11	A18/S5	
6	A10	A19/S6	
7	A9	SS0	( HIGH )
8	A8	MN/ MX	
9	AD7	RD	
10	AD6	HOLD	( RB/ GT0 )
11	AD5	HLDA	( RB/ GT1 )
12	AD4	WR	( LOCK )
13	AD3	IO/M	( S2 )
14	AD2	D1/R	( ST )
15	AD1	DEN	( S0 )
16	AD0	ALE	( QS0 )
17	NMI	INTA	( QS1 )
18	INTR	TEST	
19	CLK	READY	
20	GND	RESET	

El procesador usa ésta terminal para indicar que el ducto de direcciones contiene una dirección de memoria válida o una dirección de un puerto válido.

Una terminal utilizada para controlar la dirección del flujo del dato mediante una conexión externa a los reforzadores del bus de datos.

Es una terminal de entrada que al ser activada causa una interrupción tipo 2 y que es entendida al final de la ejecución de la actual instrucción.

Siempre que esta terminal es cero lógico, el procesador prepara su ducto de datos para recibir un dato de la memoria o de los dispositivos de E/S conectados al sistema.

Es una terminal en la interfaz con dispositivos que requieren que el ciclo ducto sea mayor en tiempo al ciclo ducto típico.

El procesador la usa para indicar que el ducto de datos contiene un dato válido para ser almacenado en la memoria o enviado a E/S.

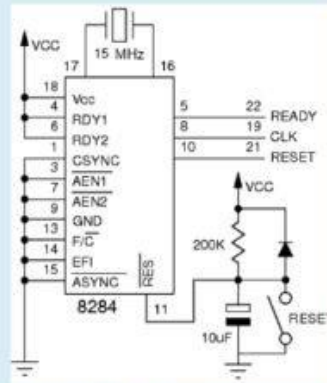
**Question 2**

Not yet answered

Marked out of 2.00

Flag question

Considerando las condiciones en las que se encuentra el generador de reloj **8284** del siguiente diagrama determine lo solicitado.



a) ¿Que frecuencia se presenta en las terminales CLK y PCLK?

CLK=  MHz PCL=  MHz

b) ¿Que frecuencia se presenta en las terminales CLK y PCLK si la terminal F/\_C se conecta a Vcc?

CLK=  PCL=

**Question 3**

Answer saved

Marked out of 2.00

Flag question

Si  $\mu P$  8088 opera con una frecuencia de **5 MHz**. ¿Cuánto tiempo se requiere para completar un ciclo ducto estándar?

ns

$$1/5 = .2 \text{ e-6}$$

200 ns

800ns

250

16MHZ

0.0625

**Question 4**

Incomplete answer

Marked out of 2.00

Flag question

Decodifique los siguientes opcode para un 8088

Dirección	Contenido
072A:100	31
072A:101	D2

Instrucción:

Registro modificado:  , valor:  h

Xor Dx,Dx

Dx = 0

**Question 5**

Not yet answered

Marked out of 2.00

Flag question

Decodifique los siguientes opcode para un 8088

Dirección Lógica	Contenido
1740:1000	E9
1740:1001	FD
1740:1002	FF

¿Qué instrucción de salto es? 

Según el contenido de memoria

¿A qué dirección lógica salta? **Nota:** Formato hexadecimal sin H

Near  
1740:1000

**Question 6**

Not yet answered

Marked out of 8.00

Flag question

Ensamble la siguiente secuencia según los mnemónicos y código máquina del uP 8088 correspondiente.

Instrucción	Código máquina (hex)
MOV AL,34h	<input type="text"/> h
MOV DX,40h	<input type="text"/> h
OUT 42h,AL	<input type="text"/> h
NEXT: IN AL,DX	<input type="text"/> h
JMP NEXT	<input type="text"/> h

Mov Bx,20h  
Mov DI,9  
Mov Dx,1  
In ax,DX  
Mov cx,1  
Next:  
Inc cx  
Out DX,AX  
loop next

B034h  
BA4000h  
E642h  
Next:EC  
EBFD



**Question 7**

Not yet answered

Marked out of 10.00

Flag question

Decodifique los siguientes opcode para un 8088 y describa utilizando los mnemónicos: (Documento de Instrucciones)

Dirección	Contenido
072A:200	B9
072A:201	10
072A:202	00
072A:203	31
072A:204	FF
072A:205	8A
072A:206	05
072A:207	E6
072A:208	42
072A:209	47
072A:20A	E2
072A:20B	F9

Y que valor tiene BX?



```
Mov cx,0010h
Xor di,di
@@next:
Mov al,[DI]
Out 42h,al
Inc di
Loop @@next
Bx = indefinido
```

🚩 Flag question



1. ¿En que modo esta trabajando el uP?
2. ¿A que frecuencia esta trabajando el uP?  MHz.
3. ¿Cuántos ciclos de espera (Tw) se insertan al trabajar con la EEPROM?
4. ¿Cuánto tiempo se requiere para completar un ciclo ducto?  ns.
5. ¿En qué memoria existen direcciones espejo? en
6. ¿Se está utilizando completamente la RAM en el sistema?
7. ¿Cuál es el rango de direcciones de la RAM en el sistema? Desde  h  
hasta la dirección  h
8. ¿Qué tamaño de memoria tiene la EEPROM?  KByte.
9. ¿Se está utilizando completamente la EPROM en el sistema?
10. ¿Cuál es el rango de direcciones de EPROM en el sistema? Desde  h  
hasta la dirección  h



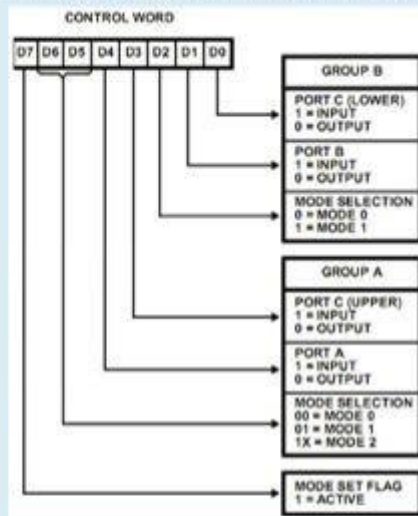
**Question 11**

Not yet answered

Marked out of 4.00

Flag question

Dado un 8255 con dirección base 80H, escriba un procedimiento lo más compacto posible para programar en modo 0 el puerto A y puerto C (bajo) de entrada, puerto B y puerto C (alto) de salida.



```
Config proc
mov al,91h
out 83h,al
endp
ret
```

**Question 12**

Not yet answered

Marked out of 4.00

Flag question

¿Qué es lo que sucede en el procesador (8088) cuando se ejecuta la INT 40h?

Se guardan las banderas

Se guarda cs y ip

Se deshabilitan las interrupciones

Se busca el nuevo cs:ip en el vector de interrupciones multiplicando 4\*numero de interrupcion

Se salta a la ISR

Con el iret recuperamos de la pila ip,cs y las banderas

hacer un corrimiento en c de un bit a la izquierda

n lecturas de 8 bits en puerto en ensamblador Write (puerto de 16 bits, bufer \* , cantidad), no tiene que ser mayor a 64kbytes

\_read\_Puerto proc

```

push bp
mov bp,sp
mov Dx,[bp+4]
mov [bx],[bp+6]
mov cx,[bp+8]
cmp cx,0
jz @@fin
@@next: in al,Dx
mov [bx],al
inc bx
loop @@next

```

```

@@fin:
Pop bp
ret
endp

```



Decodifique los siguientes opcode para un 8088 y describa utilizando los mnemónicos

Dirección	Contenido
072A:100	BE
072A:101	FF
072A:102	0F
072A:103	E5
072A:104	27
072A:105	89
072A:106	04
072A:107	4E
072A:108	75
072A:109	F9

```
Mov si,0FFFh
@@next:
In ax,27h
Mov [si],ax
Dec si
Jnz @@next
```

Diseñe e implemente una función en **ensamblador** (para ser llamada desde lenguaje C) que active un determinado bit de un determinado puerto del Atmega1280/2560.  
El puerto y el número de bit a operar son pasados como parámetros -- según la función anterior:

```
Setbitport:
Movw Z,R24
LD R26,Z ;in

ldi R28,1

ciclo: cpi R22,0
Breq fin
Lsl R28
Dec R22
Jmp ciclo

Fin:
Or R26,R28
ST Z,R26
ret
```



**Question 13**

Not yet answered

Marked out of 4.00

Flag question

Diseñe e implemente un procedimiento en **lenguaje ensamblador** (x86 -16bits) para leer escribir un byte a un puerto. El procedimiento será invocado desde programas en **lenguaje C** considerando la siguiente función prototipo:

```
void outportb( uint16_t puerto, uint8_t dato );
```

Respuesta:

```

PROC                                ; comentarios
    ;                                ; protocolo de entrada a función
    ;                                ;
    ;                                ;
    ;                                ; copiar parámetro dato
    ;                                ; copiar parámetro puerto
    ;                                ; sacar dato por puerto
    ;                                ; protocolo de salida de función
    ;                                ;
    ;                                ;
ENDP

-000-
```

```
Void outportb(uint16_t puerto, uint8_t dato)
```

```
    Push bp
```

```
    Mov bp,sp
```

```
    Mov Dx,[bp+4]
```

```
    Mov al,[bp+6]
```

```
    Out Dx,al
```

```
    Pop bp
```

```
    ret
```

```
endp
```

```
// uint8_t peek(uint16_t segmento, uint16_t desplazamiento)
```

```
_Peek proc //trae dato
```

```
    Push bp
```

```
    Mov bp,sp
```

```
    Push Ds
```

```
    Mov Ds,[bp+4]
```

```
    Mov bx,[bp+6]
```

```
    Mov al,[bx]
```

```
    Pop Ds
    Pop bp
    Ret
endp
```

```
void poke(uint16_t segment, uint16_t desp, uint8_t dato);
```

```
_poke proc
Push bp
Mov bp,sp
Push Ds
Mov Ds,[bp+4]
Mov bx,[bp+6]
Mov al,[bp+8]
Mov [bx],al
```

```
Pop Ds
Pop bp
Ret
endp
```

Question 14

Answer saved

Marked out of 4.00

Flag question

Diseñe e Implemente un procedimiento en **lenguaje ensamblador** (x86 -16bits) para leer un byte de un lugar de memoria dado el segmento y desplazamiento correspondiente (dirección lógica). El procedimiento será invocado desde programas en **lenguaje C** considerando la siguiente función prototipo:

```
uint8_t peekb ( uint16_t segm, uint16_t desp );
```

Respuesta:

```

PROC                                ; comentarios
;                                     ; protocolo de entrada a función
;                                     ;
;                                     ;
;                                     ; copiar parámetro desp al registro correspondiente
;                                     ;
;                                     ; salvar segmento original
;                                     ; copiar parámetro segm al registro correspondiente
mov     [di], [di]                  ; leer valor de memoria al registro de retorno
;                                     ; recuperar segmento original
;                                     ; protocolo de salida de función
;                                     ;
ENDP
```

-oOo-



**Question 15**

Not yet answered

Marked out of 8.00

Flag question

**Parte a)**

Diseñe un procedimiento en **lenguaje ensamblador** (para después ser llamada desde lenguaje C) para escribir un bit de un determinado dato el cual será pasado como argumento por referencia, así como el número del bit a operar y el valor del bit a escribir.

**Considere:**

```
typedef unsigned char uint8_t;
/*
 * Prototipo:
 * void WriteBit ( uint8_t *dato, uint8_t num_bit, uint8_t valor_bit );
 *
 * Ejemplo de uso:
 * Si existe uint8_t valorA=0x07
 * Escribir 1 al bit 3 de valorA
 */
WriteBit( &valorA, 3, 1 ); /* Resultado: valorA = 0x0F */
```

**Respuesta:**

A
B
I
A
E
L
S
P
U
B
C

**Question 16**

Not yet answered

Marked out of 8.00

Flag question

**Parte b)**

Diseñe una función en **lenguaje C** para desactivar un determinado bit de un determinado puerto el cual será pasado como parámetro así como el número del bit a operar.

**Notas:**

- Debe hacer uso de la técnica leer-modificar-escribir (read-modify-write)
- Deberá hacer uso de la función **WriteBit()** -- ver parte a)
- Puede hacer uso de las funciones:

```
uint8_t inportb( uint16_t puerto );
outportb ( uint16_t puerto , uint8_t dato );

typedef unsigned char uint8_t;
typedef unsigned int uint16_t;
/* Prototipo:
 * void ClrBitPort ( uint16_t puerto, uint8_t nbit );
 *
 */
ClrBitPort( 0x80, 1 );
```

**Respuesta:**

```
void ClrBitPort ( uint16_t puerto , uint8_t nbit ) {
    uint8_t x;
     = inportb( puerto ); /* leer */
     ; /* modificar */
    outportb(  ,  ); /* escribir */
}
```

```
X = inportb(puerto);
WriteBitPort(&x,nbit,0);
Ouportb(puerto,x);
```

## ENSAMBLADOR

```
ClrBitport proc
Push bp
Mov bp,sp
Mov Dx,[bp+4]
Mov cl,[bp+6]
In al,Dx
Mov ah,0feh
Rol ah,cl
And al,ah
Out Dx,al
```

```
Pop bp
ret
endp
```

### Question 17

Not yet answered

Marked out of 5.00

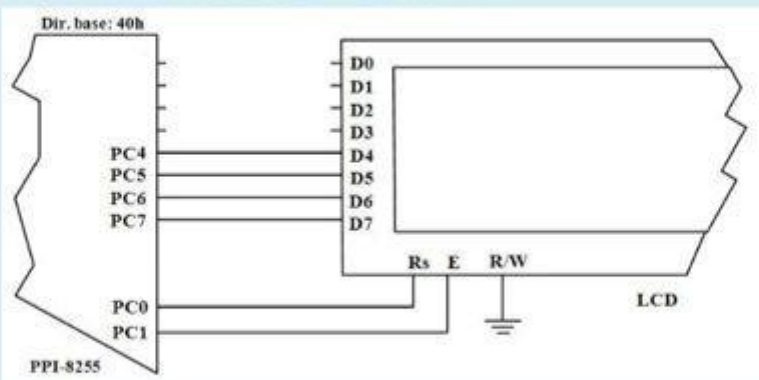
Flag question

Considerando la figura, diseñe una función en lenguaje C (WriteLCD) para escribir un dato o instrucción (Rs=1: Dato, Rs=0: Instrucción) de 8 bits al LCD. El dato a introducir al LCD se debe hacer por nibble (primero el menos significativo) y **se graba en el flanco de bajada de E**.

**Nota1:** asumir que el 8255 fue previamente programado con PC como salidas.

**Nota2:** Considere la existencia de las funciones inportb, outportb, SetBitPort, ClrBitPort, NotBitPort.

```
/* Prototipo: void WriteLCD( uint8_t tipo, uint8_t dato);
Si tipo=0 : Instrucción de otra manera dato */
```



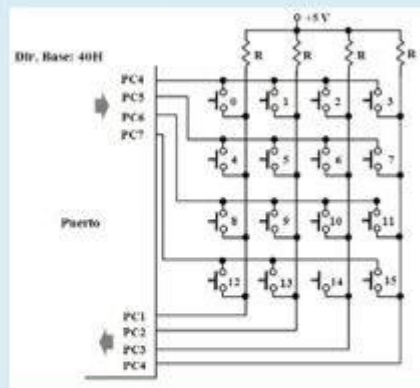
**Question 18**

Not yet answered

Marked out of 5.00

Flag question

Según la figura diseñe una función en **lenguaje C** que rastrea y determina que tecla ha sido oprimida. Si no se detecta tecla presionada la función retorna -1, de lo contrario retorna el número de la tecla presionada.



**Nota1:** Considere que la disponibilidad de las funciones **inportb**, **outputb**, **SetBitPort**, **ClrBitPort**, **NotBitPort** y **GetBitPort**

**Nota2:** Considere el **PPI-8255** ya configurado e iniciado de la manera correcta para esta aplicación específica.



```
#define PC 0x42

char Tecla_pressed(){
    char i=0,j=4,tecla = -1;
    while((tecla == -1) && (i<4) ){
        if((Getbitport(PC,i)) == 0) { //presionado
            while(j<8){
                setBitport(PC,j);
                if(GetBitport(PC,i)){
                    tecla = ((j-4)*4+i);
                    j = 8;
                }
                j++;
            }
        }
        i++;
    }
    outportb(PC, inportb(PC) & 0x0f );
    return tecla;
}
```

**Question 1**

Not yet answered

Marked out of 1.00

Flag question

¿Qué es un microcontrolador?



Un microcontrolador es un circuito integrado compacto diseñado para controlar una operación específica en un sistema embebido. Un microcontrolador típico incluye un procesador, memoria y periféricos de entrada / salida (E / S) en un solo chip

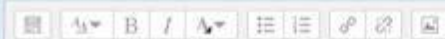
**Question 2**

Not yet answered

Marked out of 1.00

Flag question

¿Qué tiene en común un uC y un uP?



**Question 3**

Not yet answered

Marked out of 1.00

Flag question

¿En qué difiere un microcontrolador de un microprocesador?



el uC tiene puertos, ram y rom interna y el uP no por si solo.

**Question 4**

Not yet answered

Marked out of 1.00

Flag question

Indique cuales son las características de un microcontrolador:

- a) Contiene un microprocesador adentro
- b) Tiene RAM interna
- c) La mayoría manejan un conjunto de instrucciones complejas
- d) Normalmente se implementan para aplicaciones específicas
- e) Están contenidos en un solo IC

Question 5  
Not yet answered  
Marked out of 1.00  
Flag question

El microcontrolador se tarda  un ciclo de reloj para ejecutar una instrucción, esto se debe a su ejecución

### Casi siempre, encausada

Question 6  
Not yet answered  
Marked out of 3.00  
Flag question

Escriba la funcionalidad de 3 registros del CPU (No registros de propósito general) y que pueden ser operables por instrucciones.



SP -> stack pointer que apunta al tope de la pila

SREG -> status register donde estan las banderas de estado de una operacion

IP -> se encarga de traer la instruccion siguiente a ser ejecutada

Question 7  
Not yet answered  
Marked out of 1.00  
Flag question

¿Cuál es el rango de direcciones de la RAM del Microcontrolador atmega1280?

Desde la dirección 0x  hasta la dirección 0x

**0200h -21FFh**

Question 8  
Not yet answered  
Marked out of 1.00  
Flag question

¿Cuál es el rango de voltaje de entrada en un puerto digital del Microcontrolador atmega1280/2560 alimentado por una fuente de 5V DC?

Entrada nivel lógico en bajo: Desde  V hasta  V

Entrada nivel lógico en alto: Desde  V hasta  V

bajo = -0.5 hasta  $V_{CC} * 0.3$

alto =  $0.6 * V_{CC}$  hasta  $V_{CC} + 0.5$

**-0.5 a 1.5**

**3 a 5.5**

Question 9  
Not yet answered  
Marked out of 1.00  
Flag question

El Microcontrolador ATmega1280/2560 opera sobre el rango de voltaje desde  V hasta  V.

**1.8 a 5.5**



**Question 10**

Not yet answered

Marked out of 8.00

Flag question

Considerando que el procesador **no cuenta con las instrucciones de rotación (sin acarreo)** y solo tiene corrimientos a la izquierda y a la derecha. Escriba una **secuencia** para hacer la rotación (**no través del acarreo**) sobre el registro **Z**.

```
Ldi r24,80
And r24,r30
Ldi R25,80
And r25,r31
```

```
lsl r31 ;dejamos en cero la primera posición
lsl r30
Cpi r24,0 ;hay un 1 en r30?
Breq rol2
Ori r31,1
```

```
rol2:
cpi r25,0 ;hay 1 en r31?
Breq fin
Ori r30,1
```

Fin:

**Versión corta**

```
Mov r25,r30
Mov r26,r31
```

```
Lsl r30
Lsl r31
Sbrc r25,7 ;r30 es 0
Ori r31,1
Sbrc r26 ;7 es 0
Ori r30,1
```

### Question 11

Not yet answered

Marked out of 10.00

Flag question

Diseñe e implemente una función en **ensamblador** (para ser llamada desde lenguaje C) que active un determinado bit de un determinado puerto del Atmega1280/2560. El punto y el número de bit a operar son pasados como parámetros

```
typedef unsigned char uint8_t;
typedef unsigned int uint16_t;
/* Prototipo:
void SetBitPort ( uint16_t puerto, uint8_t nbit );
*/
SetBitPort( 0x80, 1 );
```



```
#define portH r25
#define portL r24
#define num_bit r22
#define mask r23
#define temp r21

movw ZL, portL
ldi mask, 0x01

tst num_bit
breq end_nxt

nxt:
    lsl mask
    dec num_bit
    brne nxt
end_nxt:
    ld temp, Z; read
    or temp, mask; modify
    st Z, temp; write
```

### Question 12

Not yet answered

Marked out of 8.00

Flag question

Considere la siguiente secuencia

```
;**** inicio ****; num. de ciclos
ldi r25,N        ; 1
next: nop         ; 1
    dec r25       ; 1
    brne next     ; 2 y 1
;**** fin ****
```

a. ¿Qué valor debe ser **N** para lograr que la secuencia consuma **768 ciclos** que en tiempo equivale a 48us si el uC ATmega2560 opera a 16MHz?

192

b. ¿Qué valor debe ser **N** para lograr que la secuencia consuma **1024 ciclos** que en tiempo equivale a 64us si el uC ATmega2560 opera a 16MHz?

0

$$4n = 768$$

0

### Question 13

Not yet answered

Marked out of 8.00

Flag question

Implemente una secuencia (repetitiva) en lenguaje ensamblador para el microcontrolador **ATmega2560** que consuma **4096 ciclos**.



**Question 14**

Not yet answered

Marked out of 10.00

Flag question

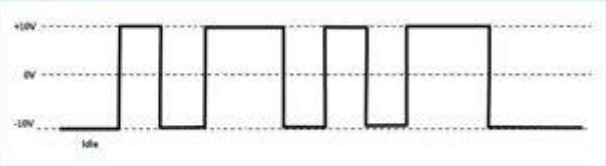
Implemente una función de retardo llamada `Delay750uS_int()` la cual se basa en el temporizador 0 (`Timer0`) para un retardo de **750uS** considerando los siguientes requerimientos.

1. La función es para un microcontrolador **Atmega1280/2560**.
2. La función deberá utilizar la(s) interrupción(es) del **timer** correspondiente.
3. El uso de **timer** es exclusivo para la función -- no existe otro código que que hace uso del timer.
4. El microcontrolador opera tiene como reloj principal el oscilador externo de **16 MHz**.
5. La forma de uso es:  
  
**Delay750uS\_int();** // va a la función y regresa logrando un retardo de 750uS //
6. La función deberá hacer uso de rutina(s) de servicio de interrupción (**RSI**) lo más compacta posible.
7. El código de la función realiza la configuración y uso de los recursos requeridos para lograr la funcionalidad.

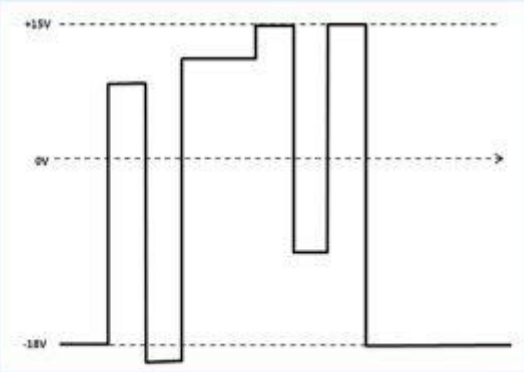
Question 15  
Not yet answered  
Marked out of 1.00  
Flag question

Considerando una comunicación UART 9600,8,N,1 y la captura del Frame de un dato que se muestra diga que dato ha sido transmitido.

a) Dato=  h




b) Dato=  h



D1  
29

Question 16  
Not yet answered  
Marked out of 2.00  
Flag question

¿Qué registros funcionan para configurar la velocidad de operación del puerto serie 01?



UCSR0A – U2X0  
UBRR0





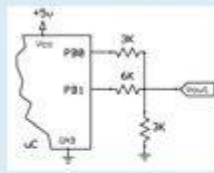
Question 19

Not yet answered

Marked out of 4.00

1" Flag question

Considerando el siguiente diagrama donde el sistema es alimentado con 5.0 Volt y las terminales PB0 y PB1 están configuradas como salida.



¿Qué voltaje está presente en Vout según la combinación de PB1 y PB0?

PB1	PB0	Vout (Volts)
0	0	<input type="text"/>
1	0	<input type="text"/>
0	1	<input type="text"/>
1	1	<input type="text"/>

0	0	0
flotado	1	2.5
0	1	2
1	flotado	1.6
1	0	1
1	1	3

Un sistema hace N lecturas de 8 bits de un puerto y las almacena en un arreglo máximo de 64K bytes. Este arreglo debe ser ordenado y se pide una función de ordenamiento en lenguaje C el cual tiene el siguiente declaración prototipo

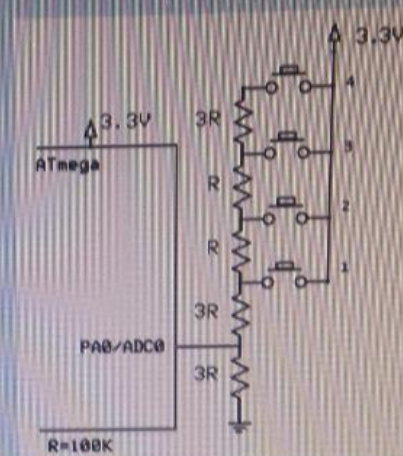
```
void OrdenaDatos ( uint8_t *buffer, uint16_t Cantidad );
```

Implemente el procedimiento haciendo uso de un algoritmo de ordenación básico.

Implementación:

```
Void OrdenaDatos(uint8_t *buffer, uint16_t cantidad){
    Uint16_t i,j;
    Uint8_t aux;
    For(i=0; i<cantidad-1; i++){
        For(j=i+1; j<cantidad; j++){
            If(Buffer[i]>Buffer[j]){
                Aux = Buffer[i];
                Buffer[i] = buffer[j];
                Buffer[j] = aux;
            }
        }
    }
}
```

Diseñe e implemente las siguientes funciones para detectar la tecla presionada según el diagrama



- Función `Init_ReadKeys()` para inicializar recursos según aplicación.
- Función `ReadKeys()` para determinar que tecla se presionó

Calcule el número de ciclos requeridos para la ejecución del siguiente procedimiento (no incluya los ciclos de llamada al procedimiento)

```
Test: eor    r24,r24
Inc   r24
Lsl   r24
Ori   r24,1
Brcc  PC-2
Ret
```

Corrimiento con signo

`_corrimientoSigno` proc

Push bp

Mov ax,[bp+4]

Sar ax,1

Pop bp

Ret

endp

Considerando que el microcontrolador ATmega1280/2560 no cuenta con las instrucciones de rotación (sin acarreo) y solo tiene comentarios a la izquierda y a la derecha. Escriba una secuencia para hacer la rotación (no través del acarreo) sobre el registro R24 para el caso de rotación a la derecha.

	; comentarios
	; borrar acarreo
	; verificar el bit correspondiente
sec	; activar acarreo si el bit esta activo
lsl R24	; corrimiento del registro

Considere la siguiente declaración de tipo de dato implemente en lenguaje C las 3 funciones prototipo que operan ese tipo de dato donde N es un valor entero.

```
#define BUFFER_SIZE N

typedef struct {
    char buffer[BUFFER_SIZE]; /* espacio reservado */
    volatile unsigned char in; /* indice entrada (head) */
    volatile unsigned char out; /* indice entrada (tail) */
} ring_buffer_t;

void insertar( ring_buffer_t *cola, char dato );
unsigned char available( ring_buffer_t *cola );
char remover( ring_buffer_t *cola );
```

Haciendo uso de la función `getnibble` escriba una función llamada `UART0_sendDataHex` para desplegar en la terminal (via UART0) en formato hexadecimal el nibble (menos significativo), el byte (menos significativo) o la palabra recibe como parámetro. El tamaño del dato a desplegar es seleccionado según el valor del parámetro tam donde 1 es para desplegar el nibble, 2 desplegar el byte y 4 desplegar la palabra. **Nota:** si la función recibe un valor de tamaño incorrecto, no envía dato alguno a la terminal.

```
void UART0_sendDataHex( uint16_t dato, uint8_t tam);
```

Forma de uso (secuencia de ejemplo)

```
UART0_sendDataHex(0x12AB,4); /* se presenta en la terminal la palabra completa 12AB */
UART0_sendDataHex(0x12AB,2); /* se presenta en la terminal el byte menos significativo AB */
UART0_sendDataHex(0x12AB,1); /* se presenta en la terminal el nibble menos significativo B */
```

Diseñe e implemente una función compacto en lenguaje C para retornar en ASCII el nibble menos significativo un byte que recibe como parámetro.

```
char getnibble( uint8_t dato );
```

Ejemplos:

```
x = getnibble( 0x3F); x = 'F'
x = getnibble( 0x38); x = '8'
```

```
char getnibble(uint8_t dato){
    return (dato&0f);
}
```

Escriba la secuencia en lenguaje ensamblador del ATmega1280/2560 para sumar dos número de 16 bits apuntados por los registros X y Z -- El resultado deberá estar en el lugar apuntado por X.

```
movw r18,x

ld r20,x+
ld r21,x
ld r22,z
ld r23,z+1

add r20,r22
adc r21,r23

movw x,r18
ST x+,r20
ST x,r21
movw x,r18
```

Considerando que el uC opera a 16MHz, escribe la secuencia para configurar el UART1 como 15K,8,E,2

Implemente una función llamada Delay20us, la cual se basa en el timer0, “no usar interrupciones”, reloj interno a 16MHz

```
Delay20US() {
    TCNT0 = 0;
    OCR0A = 4;
    TCCR0A = 2;
    TCCR0B = (3<<CS00);

    while(!((TIFR0>>OCF0A) & 1));
    TIFR0 = (1<<OCF0A);
    TCCR0B = 0;
}
```

Cuales son las restricciones que existen en las instrucciones SBI y CBI?

Opera solo en los 32 registros E/S y de r0 a r31

Escribe una secuencia corta en lenguaje ensamblador o C para escribir ceros en 0x00 en toda la memoria SRAM de microcontrolador atmega1280

Considera que el procesador no cuenta con las instrucciones de rotación y solo tiene corrimientos, escriba una secuencia para hacer la rotación sobre el r24, para el caso de rotación a la derecha

```
mov R25,R24
lsr R24
sbrc R25,0
ori R24,0x80
```

Cual es la dirección de memoria donde inicia la pila del microcontrolador atmega 1280?  
Dir Registro: 0x3E – Dir inicial Pila: 21FF

Escriba la secuencia para programar el bit 2 del puerto "C" en 3er estado  
DDRC &=(~ (1<<PC2));  
PORTC &= (~(1<<PC2));

Cual es la dirección indirecta del contador de programa?

Diseñe e implemente un procedimiento en lenguaje "C" para desplegar (via puerto serie 0) un byte en formato hexadecimal. Void UART\_putByteHexa(uint8\_t dato)

```
void UART_putByteHexa(uint8_t dato);
uint8_t i,x=4;
for(i=0; i<2; i++){
    if((dato>>x)&0f < 10)
        UART0_putchar(((dato>>x)&0f) + 0x30);
    else
        UART0_putchar(((dato>>4)&0f) + 0x37);
    x-=4;
}
```

Implemente la función EPROM\_write32bits(), la cual escribe un dato de 32 bits en una dirección dada de la eeprom. Respetar el formato big endian

Imprimir en binario 32bits por UART

```
32bitsPorUART(uint32_t dato, uint16_t nDatos){
    if(nDatos < 33 && nDatos != 0){
        uint32_t mask = (1<<(nDatos-1));
        for(i=0; i<nDatos; i++){
            putchar_UART((mask>>i)&dato)? 1:0;
        }
    }
}
```

Cual es la dirección indirecta del apuntador de pila

Lista 3 ventajas y desventajas de los microcontroladores

Por que se dice que la arquitectura del microcontrolador atmega de la familia avr es "RISC-Like"

Por que existen instrucciones con diferente cantidad de ciclos maquina requeridos para su ejecución?

Escriba la secuencia para configurar el USART: 19200,8,N,1

UCSR0A = 0x2;

UCSR0B = 0x18;

UCSR0C = 0x06;

UBRR0 = 103

Error = 0.16

Escriba una secuencia corta de instrucciones en ensamblador para escribir 0xFF en todos los registros de propósito general

```
ldi R31,0x0
ldi R30,0x0
ldi R29,0xff
```

```
next: ST Z+,R29
cpi R30,0x1D
brne next
```

```
movw Z,R28
```

Diseñe e implemente un procedimiento en lenguaje ensamblador (x86) para leer/escribir un byte a un puerto. El procedimiento será invocado desde programas en lenguaje C. Considerando la siguiente función prototipo: void outportb(uint16\_t puerto, uint8\_t dato)

```
_outportb proc
Push bp
Mov bp,sp
mov DX,[bp+4]
mov ax,[bp+6]
out Dx,ax

Pop bp
ret
endp
```

Diseñe un procedimiento en lenguaje ensamblador (para después ser llamado desde lenguaje C), para desactivar un bit de un determinado dato el cual será pasado como parámetro de referencia, así como el número de bit a operar prototipo:

void ClrBit(uint8\_t \*dato, uint8\_t nbit);

```
_ClrBit proc
Push bp
Mov bp,sp
Mov bx,[bp+4]
Mov cl,[bp+6]
Mov ah,0feh;
rol ah,cl
And [bx],ah
Pop bp
Ret
endp
```

Diseñe una función en lenguaje c para desactivar un determinado bit de un determinado puerto el cual será pasado como parámetro así como el número de bit a operar

- Debe hacer uso de las técnicas leer-modificar-escribir
- Deberá hacer uso de la función Clrbit() del inciso anterior
- Puede hacer uso de las funciones: uint8\_t inportb(uint16\_t puerto); outportb(uint16\_t puerto, uint8\_t nbit)



```

Void WriteBitport(uint16_t puerto, uint8_t n_bit){
    Uint8_t x = inportb(puerto);
    Clrbit(&x,n_bit);
    Outportb(puerto,x);
}

```

Realiza el mismo procedimiento en lenguaje ensamblador (x86)

```

_writeBitport proc
    push bp
    mov bp,sp
    mov Dx,[bp+4]
    mov cl,[bp+6]
    in al,Dx
    mov ah,0feh
    rol ah,cl
    and al,ah
    out Dx,al

    pop bp
    ret
endp

```

Dado que cualquier puerto del 8255 una vez programado como salida (modo 0), este puede ser leído para conocer su estado actual, diseñe las siguientes funciones

Nota: Puede hacer uso de las funciones inportb y outportb

Nota2: #define unsigned char BYTE y #define unsigned int WORD

- A) Función para leer cualquier bit de un puerto del 8255 y retornar su valor (1 o 0)
- BYTE GetBitPort(WORD puerto, BYTE num\_bit);

- B) Función para leer y desactivar cualquier bit de un puerto del 8255
- Void WriteBitPort(WORD puerto, BYTE num\_bit, BYTE valor\_bit)

Decodifique los siguientes opcode para un 8088

Dirección lógica	Contenido
072A : 100	C6
072A : 101	02
072A : 102	37

Ensamble los siguientes mnemónicos a código máquina del uP 8088

Instrucción	Código máquina (HEX)
Mov al,34h	
Mov DX, 40h	
Out 42h, al	
In al, dx	

Decodifique los siguientes opcode para un 8088 y descríbalos utilizando mnemónicos

Dirección	Contenido	mnemónicos
072A : 100	B9	
072A : 101	10	
072A : 102	00	
072A : 103	31	
072A : 104	DB	
072A : 105	E4	
072A : 106	40	
072A : 107	88	
072A : 108	07	
072A : 109	43	
072A : 10 <sup>a</sup>	E2	
072A : 10B	F9	

¿Qué valor tiene BX?

Configura un uart genérico con paridad Even (15k,8,E,2)