SISTEMAS BASADOS EN MICROPROCESADOR **Segunda convocatoria 2023/2024**

Apellidos:.....

Nombre:.....

P1	P2	P3

Duración 4:00 horas

1.- [3 puntos] Sea el siguiente código:

```
;                               Emulación       CiclosCódigo máquina
;-------------------------------------------------------------------------------
QuienSabe   clr.w  r14          ;_____________|___|_____________________________
            mov.w  #8, r15      ;_____________|___|_____________________________
bucle       rla.b  r12          ;_____________|___|_____________________________
            adc.w  r14          ;_____________|___|_____________________________
            dec.w  r15          ;_____________|___|_____________________________
            jnz    bucle        ;_____________|___|_____________________________
            mov.w  r14, r12     ;_____________|___|_____________________________
            ret                 ;_____________|___|_____________________________
```

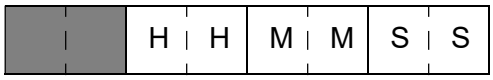
a) Analícelo en indique qué hace. Señale si sigue o no el convenio de llamada de C. En caso afirmativo, muestre el prototipo. En caso negativo, enumere los parámetros de entrada/ salida, ubicación y tipo.

b) Calcule el tiempo total de ejecución para F_{MCLK}=1MHz.

c) Ensamble manualmente el programa. ¿Cuántas palabras ocupa el código?

2.- [3 puntos] Se desea realizar la función `IncTime` que recibe una información de hora y la incrementa en un segundo:

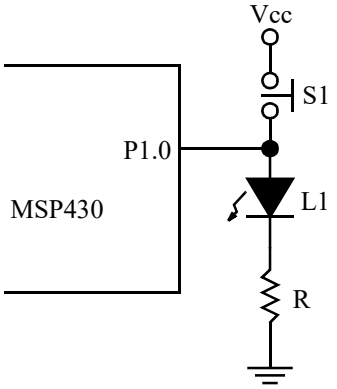
```
int IncTime (unsigned long *hms);
```



`hms` es un puntero a un entero largo sin signo que almacena, en BCD empaquetado, información temporal en formato HHMMSS (dos dígitos BCD para hora, minutos y segundos):

La función sigue el convenio de llamada de C y devuelve 1 si se produce un cambio de día (la hora de entrada es 23:59:59), -1 si la hora es incorrecta y 0 en el resto de los casos. Ignore el byte alto de la información horaria (zona rayada). La hora se guarda en formato de 24h. Escriba `IncTime` en ensamblador del MSP430. NOTA: La nueva hora se guarda en el mismo sitio que la original. Se recomienda el uso de las instrucciones que soporten aritmética BCD.

3.- [4 puntos] Observe el circuito de la figura. Inicialmente el puerto se configurará en modo entrada para poder leer el estado del pulsador. Cuando el usuario lo acciona, se enciende el led sin intervención del MSP. Se desea hacer un programa que mantenga el led encendido 1 segundo más desde la liberación del pulsador. Una vez que pase ese tiempo, se volverá al estado inicial. El programa en ensamblador debe funcionar totalmente por interrupciones, con ayuda del Timer A0 y operar en el modo de mayor bajo consumo. Considere los pulsadores libres de rebotes, el perro guardián desactivado, los puertos desbloqueados, pila inicializada, ACLK=LFXT en marcha con cristal de 32768Hz y TA0.CCR0 como función primaria del P1.0, CCI0A cuando es entrada y OUT0 cuando es salida.



SISTEMAS BASADOS EN MICROPROCESADOR **Segunda convocatoria 2023/2024**

Apellidos:.....

Nombre:.....

P1	P2	P3

Duración 4:00 horas

1.- [3 puntos] Sea el siguiente código:

```
;                               Emulación       CiclosCódigo máquina
;-------------------------------------------------------------------------------
QuienSabe   clr.w  r14          ;_____________|___|_____________________________
            mov.w  #8, r15      ;_____________|___|_____________________________
bucle       rla.b  r12          ;_____________|___|_____________________________
            adc.w  r14          ;_____________|___|_____________________________
            dec.w  r15          ;_____________|___|_____________________________
            jnz    bucle        ;_____________|___|_____________________________
            mov.w  r14, r12     ;_____________|___|_____________________________
            ret                 ;_____________|___|_____________________________
```

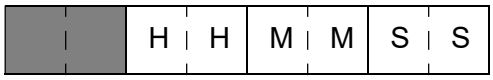
a) Analícelo en indique qué hace. Señale si sigue o no el convenio de llamada de C. En caso afirmativo, muestre el prototipo. En caso negativo, enumere los parámetros de entrada/ salida, ubicación y tipo.

b) Calcule el tiempo total de ejecución para F_{MCLK}=1MHz.

c) Ensamble manualmente el programa. ¿Cuántas palabras ocupa el código?

2.- [3 puntos] Se desea realizar la función `IncTime` que recibe una información de hora y la incrementa en un segundo:

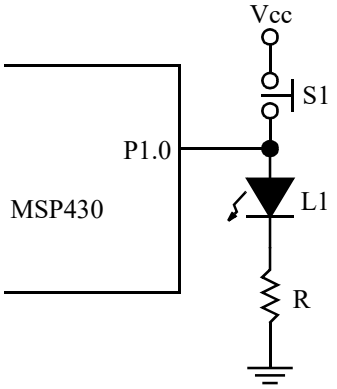
```
int IncTime (unsigned long *hms);
```



`hms` es un puntero a un entero largo sin signo que almacena, en BCD empaquetado, información temporal en formato HHMMSS (dos dígitos BCD para hora, minutos y segundos):

La función sigue el convenio de llamada de C y devuelve 1 si se produce un cambio de día (la hora de entrada es 23:59:59), -1 si la hora es incorrecta y 0 en el resto de los casos. Ignore el byte alto de la información horaria (zona rayada). La hora se guarda en formato de 24h. Escriba `IncTime` en ensamblador del MSP430. NOTA: La nueva hora se guarda en el mismo sitio que la original. Se recomienda el uso de las instrucciones que soporten aritmética BCD.

3.- [4 puntos] Observe el circuito de la figura. Inicialmente el puerto se configurará en modo entrada para poder leer el estado del pulsador. Cuando el usuario lo acciona, se enciende el led sin intervención del MSP. Se desea hacer un programa que mantenga el led encendido 1 segundo más desde la liberación del pulsador. Una vez que pase ese tiempo, se volverá al estado inicial. El programa en ensamblador debe funcionar totalmente por interrupciones, con ayuda del Timer A0 y operar en el modo de mayor bajo consumo. Considere los pulsadores libres de rebotes, el perro guardián desactivado, los puertos desbloqueados, pila inicializada, ACLK=LFXT en marcha con cristal de 32768Hz y TA0.CCR0 como función primaria del P1.0, CCI0A cuando es entrada y OUT0 cuando es salida.



SISTEMAS BASADOS EN MICROPROCESADOR **Segunda convocatoria 2023/2024**

Apellidos:.....

Nombre:.....

P1	P2	P3

Duración 4:00 horas

1.- [3 puntos] Sea el siguiente código:

```
;                               Emulación       CiclosCódigo máquina
;-------------------------------------------------------------------------------
QuienSabe   clr.w  r14          ;_____________|___|_____________________________
            mov.w  #8, r15      ;_____________|___|_____________________________
bucle       rla.b  r12          ;_____________|___|_____________________________
            adc.w  r14          ;_____________|___|_____________________________
            dec.w  r15          ;_____________|___|_____________________________
            jnz    bucle        ;_____________|___|_____________________________
            mov.w  r14, r12     ;_____________|___|_____________________________
            ret                 ;_____________|___|_____________________________
```

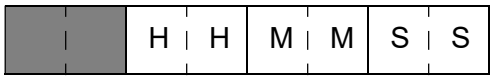
a) Analícelo en indique qué hace. Señale si sigue o no el convenio de llamada de C. En caso afirmativo, muestre el prototipo. En caso negativo, enumere los parámetros de entrada/ salida, ubicación y tipo.

b) Calcule el tiempo total de ejecución para F_{MCLK}=1MHz.

c) Ensamble manualmente el programa. ¿Cuántas palabras ocupa el código?

2.- [3 puntos] Se desea realizar la función `IncTime` que recibe una información de hora y la incrementa en un segundo:

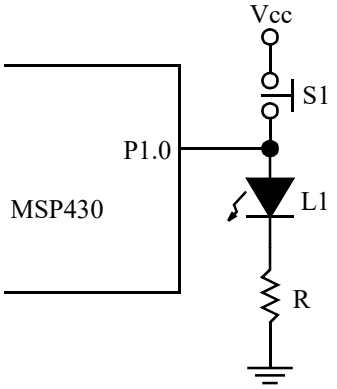
```
int IncTime (unsigned long *hms);
```



`hms` es un puntero a un entero largo sin signo que almacena, en BCD empaquetado, información temporal en formato HHMMSS (dos dígitos BCD para hora, minutos y segundos):

La función sigue el convenio de llamada de C y devuelve 1 si se produce un cambio de día (la hora de entrada es 23:59:59), -1 si la hora es incorrecta y 0 en el resto de los casos. Ignore el byte alto de la información horaria (zona rayada). La hora se guarda en formato de 24h. Escriba `IncTime` en ensamblador del MSP430. NOTA: La nueva hora se guarda en el mismo sitio que la original. Se recomienda el uso de las instrucciones que soporten aritmética BCD.

3.- [4 puntos] Observe el circuito de la figura. Inicialmente el puerto se configurará en modo entrada para poder leer el estado del pulsador. Cuando el usuario lo acciona, se enciende el led sin intervención del MSP. Se desea hacer un programa que mantenga el led encendido 1 segundo más desde la liberación del pulsador. Una vez que pase ese tiempo, se volverá al estado inicial. El programa en ensamblador debe funcionar totalmente por interrupciones, con ayuda del Timer A0 y operar en el modo de mayor bajo consumo. Considere los pulsadores libres de rebotes, el perro guardián desactivado, los puertos desbloqueados, pila inicializada, ACLK=LFXT en marcha con cristal de 32768Hz y TA0.CCR0 como función primaria del P1.0, CCI0A cuando es entrada y OUT0 cuando es salida.



SISTEMAS BASADOS EN MICROPROCESADOR **Segunda convocatoria 2023/2024**

Apellidos:.....

Nombre:.....

P1	P2	P3

Duración 4:00 horas

1.- [3 puntos] Sea el siguiente código:

```
;                               Emulación       CiclosCódigo máquina
;-------------------------------------------------------------------------------
QuienSabe   clr.w  r14          ;_____________|___|_____________________________
            mov.w  #8, r15      ;_____________|___|_____________________________
bucle       rla.b  r12          ;_____________|___|_____________________________
            adc.w  r14          ;_____________|___|_____________________________
            dec.w  r15          ;_____________|___|_____________________________
            jnz    bucle        ;_____________|___|_____________________________
            mov.w  r14, r12     ;_____________|___|_____________________________
            ret                 ;_____________|___|_____________________________
```

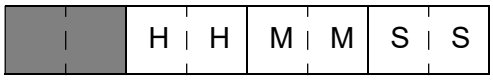
a) Analícelo en indique qué hace. Señale si sigue o no el convenio de llamada de C. En caso afirmativo, muestre el prototipo. En caso negativo, enumere los parámetros de entrada/ salida, ubicación y tipo.

b) Calcule el tiempo total de ejecución para F_{MCLK}=1MHz.

c) Ensamble manualmente el programa. ¿Cuántas palabras ocupa el código?

2.- [3 puntos] Se desea realizar la función `IncTime` que recibe una información de hora y la incrementa en un segundo:

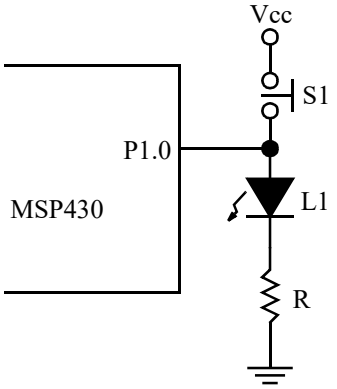
```
int IncTime (unsigned long *hms);
```



`hms` es un puntero a un entero largo sin signo que almacena, en BCD empaquetado, información temporal en formato HHMMSS (dos dígitos BCD para hora, minutos y segundos):

La función sigue el convenio de llamada de C y devuelve 1 si se produce un cambio de día (la hora de entrada es 23:59:59), -1 si la hora es incorrecta y 0 en el resto de los casos. Ignore el byte alto de la información horaria (zona rayada). La hora se guarda en formato de 24h. Escriba `IncTime` en ensamblador del MSP430. NOTA: La nueva hora se guarda en el mismo sitio que la original. Se recomienda el uso de las instrucciones que soporten aritmética BCD.

3.- [4 puntos] Observe el circuito de la figura. Inicialmente el puerto se configurará en modo entrada para poder leer el estado del pulsador. Cuando el usuario lo acciona, se enciende el led sin intervención del MSP. Se desea hacer un programa que mantenga el led encendido 1 segundo más desde la liberación del pulsador. Una vez que pase ese tiempo, se volverá al estado inicial. El programa en ensamblador debe funcionar totalmente por interrupciones, con ayuda del Timer A0 y operar en el modo de mayor bajo consumo. Considere los pulsadores libres de rebotes, el perro guardián desactivado, los puertos desbloqueados, pila inicializada, ACLK=LFXT en marcha con cristal de 32768Hz y TA0.CCR0 como función primaria del P1.0, CCI0A cuando es entrada y OUT0 cuando es salida.



SISTEMAS BASADOS EN MICROPROCESADOR **Segunda convocatoria 2023/2024**

Apellidos:.....

Nombre:.....

P1	P2	P3

Duración 4:00 horas

1.- [3 puntos] Sea el siguiente código:

```
;                               Emulación       CiclosCódigo máquina
;-------------------------------------------------------------------------------
QuienSabe   clr.w  r14          ;_____________|___|_____________________________
            mov.w  #8, r15      ;_____________|___|_____________________________
bucle       rla.b  r12          ;_____________|___|_____________________________
            adc.w  r14          ;_____________|___|_____________________________
            dec.w  r15          ;_____________|___|_____________________________
            jnz    bucle        ;_____________|___|_____________________________
            mov.w  r14, r12     ;_____________|___|_____________________________
            ret                 ;_____________|___|_____________________________
```

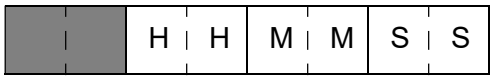
a) Analícelo en indique qué hace. Señale si sigue o no el convenio de llamada de C. En caso afirmativo, muestre el prototipo. En caso negativo, enumere los parámetros de entrada/ salida, ubicación y tipo.

b) Calcule el tiempo total de ejecución para F_{MCLK}=1MHz.

c) Ensamble manualmente el programa. ¿Cuántas palabras ocupa el código?

2.- [3 puntos] Se desea realizar la función `IncTime` que recibe una información de hora y la incrementa en un segundo:

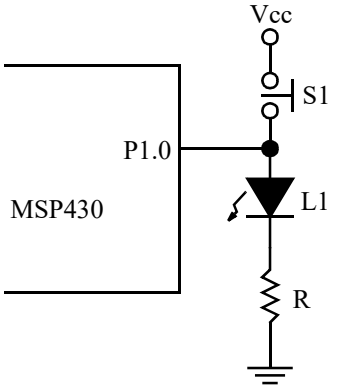
```
int IncTime (unsigned long *hms);
```



`hms` es un puntero a un entero largo sin signo que almacena, en BCD empaquetado, información temporal en formato HHMMSS (dos dígitos BCD para hora, minutos y segundos):

La función sigue el convenio de llamada de C y devuelve 1 si se produce un cambio de día (la hora de entrada es 23:59:59), -1 si la hora es incorrecta y 0 en el resto de los casos. Ignore el byte alto de la información horaria (zona rayada). La hora se guarda en formato de 24h. Escriba `IncTime` en ensamblador del MSP430. NOTA: La nueva hora se guarda en el mismo sitio que la original. Se recomienda el uso de las instrucciones que soporten aritmética BCD.

3.- [4 puntos] Observe el circuito de la figura. Inicialmente el puerto se configurará en modo entrada para poder leer el estado del pulsador. Cuando el usuario lo acciona, se enciende el led sin intervención del MSP. Se desea hacer un programa que mantenga el led encendido 1 segundo más desde la liberación del pulsador. Una vez que pase ese tiempo, se volverá al estado inicial. El programa en ensamblador debe funcionar totalmente por interrupciones, con ayuda del Timer A0 y operar en el modo de mayor bajo consumo. Considere los pulsadores libres de rebotes, el perro guardián desactivado, los puertos desbloqueados, pila inicializada, ACLK=LFXT en marcha con cristal de 32768Hz y TA0.CCR0 como función primaria del P1.0, CCI0A cuando es entrada y OUT0 cuando es salida.



SISTEMAS BASADOS EN MICROPROCESADOR **Segunda convocatoria 2023/2024**

Apellidos:.....

Nombre:.....

P1	P2	P3

Duración 4:00 horas

1.- [3 puntos] Sea el siguiente código:

```
;                               Emulación       CiclosCódigo máquina
;-------------------------------------------------------------------------------
QuienSabe   clr.w  r14          ;_____________|___|_____________________________
            mov.w  #8, r15      ;_____________|___|_____________________________
bucle       rla.b  r12          ;_____________|___|_____________________________
            adc.w  r14          ;_____________|___|_____________________________
            dec.w  r15          ;_____________|___|_____________________________
            jnz    bucle        ;_____________|___|_____________________________
            mov.w  r14, r12     ;_____________|___|_____________________________
            ret                 ;_____________|___|_____________________________
```

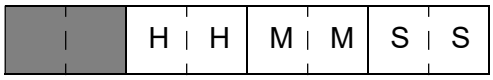
a) Analícelo en indique qué hace. Señale si sigue o no el convenio de llamada de C. En caso afirmativo, muestre el prototipo. En caso negativo, enumere los parámetros de entrada/ salida, ubicación y tipo.

b) Calcule el tiempo total de ejecución para F_{MCLK}=1MHz.

c) Ensamble manualmente el programa. ¿Cuántas palabras ocupa el código?

2.- [3 puntos] Se desea realizar la función `IncTime` que recibe una información de hora y la incrementa en un segundo:

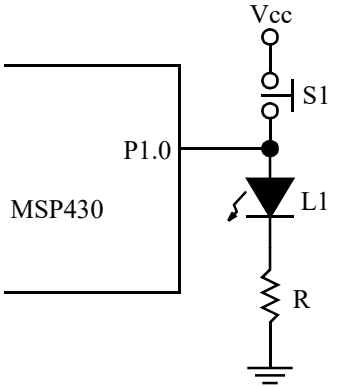
```
int IncTime (unsigned long *hms);
```



`hms` es un puntero a un entero largo sin signo que almacena, en BCD empaquetado, información temporal en formato HHMMSS (dos dígitos BCD para hora, minutos y segundos):

La función sigue el convenio de llamada de C y devuelve 1 si se produce un cambio de día (la hora de entrada es 23:59:59), -1 si la hora es incorrecta y 0 en el resto de los casos. Ignore el byte alto de la información horaria (zona rayada). La hora se guarda en formato de 24h. Escriba `IncTime` en ensamblador del MSP430. NOTA: La nueva hora se guarda en el mismo sitio que la original. Se recomienda el uso de las instrucciones que soporten aritmética BCD.

3.- [4 puntos] Observe el circuito de la figura. Inicialmente el puerto se configurará en modo entrada para poder leer el estado del pulsador. Cuando el usuario lo acciona, se enciende el led sin intervención del MSP. Se desea hacer un programa que mantenga el led encendido 1 segundo más desde la liberación del pulsador. Una vez que pase ese tiempo, se volverá al estado inicial. El programa en ensamblador debe funcionar totalmente por interrupciones, con ayuda del Timer A0 y operar en el modo de mayor bajo consumo. Considere los pulsadores libres de rebotes, el perro guardián desactivado, los puertos desbloqueados, pila inicializada, ACLK=LFXT en marcha con cristal de 32768Hz y TA0.CCR0 como función primaria del P1.0, CCI0A cuando es entrada y OUT0 cuando es salida.



SISTEMAS BASADOS EN MICROPROCESADOR **Segunda convocatoria 2023/2024**

Apellidos:.....

Nombre:.....

P1	P2	P3

Duración 4:00 horas

1.- [3 puntos] Sea el siguiente código:

```
;                               Emulación       CiclosCódigo máquina
;-------------------------------------------------------------------------------
QuienSabe   clr.w  r14          ;_____________|___|_____________________________
            mov.w  #8, r15      ;_____________|___|_____________________________
bucle       rla.b  r12          ;_____________|___|_____________________________
            adc.w  r14          ;_____________|___|_____________________________
            dec.w  r15          ;_____________|___|_____________________________
            jnz    bucle        ;_____________|___|_____________________________
            mov.w  r14, r12     ;_____________|___|_____________________________
            ret                 ;_____________|___|_____________________________
```

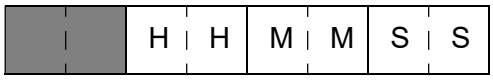
a) Analícelo en indique qué hace. Señale si sigue o no el convenio de llamada de C. En caso afirmativo, muestre el prototipo. En caso negativo, enumere los parámetros de entrada/ salida, ubicación y tipo.

b) Calcule el tiempo total de ejecución para F_{MCLK}=1MHz.

c) Ensamble manualmente el programa. ¿Cuántas palabras ocupa el código?

2.- [3 puntos] Se desea realizar la función `IncTime` que recibe una información de hora y la incrementa en un segundo:

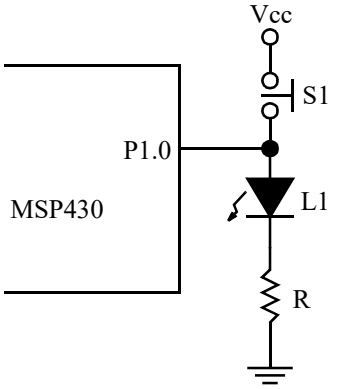
```
int IncTime (unsigned long *hms);
```



`hms` es un puntero a un entero largo sin signo que almacena, en BCD empaquetado, información temporal en formato HHMMSS (dos dígitos BCD para hora, minutos y segundos):

La función sigue el convenio de llamada de C y devuelve 1 si se produce un cambio de día (la hora de entrada es 23:59:59), -1 si la hora es incorrecta y 0 en el resto de los casos. Ignore el byte alto de la información horaria (zona rayada). La hora se guarda en formato de 24h. Escriba `IncTime` en ensamblador del MSP430. NOTA: La nueva hora se guarda en el mismo sitio que la original. Se recomienda el uso de las instrucciones que soporten aritmética BCD.

3.- [4 puntos] Observe el circuito de la figura. Inicialmente el puerto se configurará en modo entrada para poder leer el estado del pulsador. Cuando el usuario lo acciona, se enciende el led sin intervención del MSP. Se desea hacer un programa que mantenga el led encendido 1 segundo más desde la liberación del pulsador. Una vez que pase ese tiempo, se volverá al estado inicial. El programa en ensamblador debe funcionar totalmente por interrupciones, con ayuda del Timer A0 y operar en el modo de mayor bajo consumo. Considere los pulsadores libres de rebotes, el perro guardián desactivado, los puertos desbloqueados, pila inicializada, ACLK=LFXT en marcha con cristal de 32768Hz y TA0.CCR0 como función primaria del P1.0, CCI0A cuando es entrada y OUT0 cuando es salida.



SISTEMAS BASADOS EN MICROPROCESADOR **Segunda convocatoria 2023/2024**

Apellidos:.....

Nombre:.....

P1	P2	P3

Duración 4:00 horas

1.- [3 puntos] Sea el siguiente código:

```
;                               Emulación       CiclosCódigo máquina
;-------------------------------------------------------------------------------
QuienSabe   clr.w  r14          ;_____________|___|_____________________________
            mov.w  #8, r15      ;_____________|___|_____________________________
bucle       rla.b  r12          ;_____________|___|_____________________________
            adc.w  r14          ;_____________|___|_____________________________
            dec.w  r15          ;_____________|___|_____________________________
            jnz    bucle        ;_____________|___|_____________________________
            mov.w  r14, r12     ;_____________|___|_____________________________
            ret                 ;_____________|___|_____________________________
```

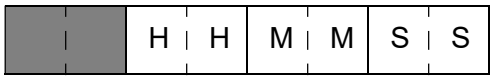
a) Analícelo en indique qué hace. Señale si sigue o no el convenio de llamada de C. En caso afirmativo, muestre el prototipo. En caso negativo, enumere los parámetros de entrada/ salida, ubicación y tipo.

b) Calcule el tiempo total de ejecución para F_{MCLK}=1MHz.

c) Ensamble manualmente el programa. ¿Cuántas palabras ocupa el código?

2.- [3 puntos] Se desea realizar la función `IncTime` que recibe una información de hora y la incrementa en un segundo:

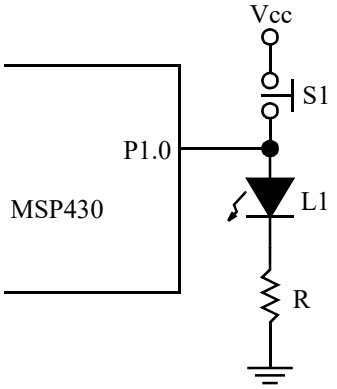
```
int IncTime (unsigned long *hms);
```



`hms` es un puntero a un entero largo sin signo que almacena, en BCD empaquetado, información temporal en formato HHMMSS (dos dígitos BCD para hora, minutos y segundos):

La función sigue el convenio de llamada de C y devuelve 1 si se produce un cambio de día (la hora de entrada es 23:59:59), -1 si la hora es incorrecta y 0 en el resto de los casos. Ignore el byte alto de la información horaria (zona rayada). La hora se guarda en formato de 24h. Escriba `IncTime` en ensamblador del MSP430. NOTA: La nueva hora se guarda en el mismo sitio que la original. Se recomienda el uso de las instrucciones que soporten aritmética BCD.

3.- [4 puntos] Observe el circuito de la figura. Inicialmente el puerto se configurará en modo entrada para poder leer el estado del pulsador. Cuando el usuario lo acciona, se enciende el led sin intervención del MSP. Se desea hacer un programa que mantenga el led encendido 1 segundo más desde la liberación del pulsador. Una vez que pase ese tiempo, se volverá al estado inicial. El programa en ensamblador debe funcionar totalmente por interrupciones, con ayuda del Timer A0 y operar en el modo de mayor bajo consumo. Considere los pulsadores libres de rebotes, el perro guardián desactivado, los puertos desbloqueados, pila inicializada, ACLK=LFXT en marcha con cristal de 32768Hz y TA0.CCR0 como función primaria del P1.0, CCI0A cuando es entrada y OUT0 cuando es salida.



SISTEMAS BASADOS EN MICROPROCESADOR **Segunda convocatoria 2023/2024**

Apellidos:.....

Nombre:.....

P1	P2	P3

Duración 4:00 horas

1.- [3 puntos] Sea el siguiente código:

```
;                               Emulación       CiclosCódigo máquina
;-------------------------------------------------------------------------------
QuienSabe   clr.w  r14          ;_____________|___|_____________________________
            mov.w  #8, r15      ;_____________|___|_____________________________
bucle       rla.b  r12          ;_____________|___|_____________________________
            adc.w  r14          ;_____________|___|_____________________________
            dec.w  r15          ;_____________|___|_____________________________
            jnz    bucle        ;_____________|___|_____________________________
            mov.w  r14, r12     ;_____________|___|_____________________________
            ret                 ;_____________|___|_____________________________
```

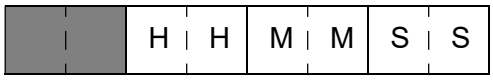
a) Analícelo en indique qué hace. Señale si sigue o no el convenio de llamada de C. En caso afirmativo, muestre el prototipo. En caso negativo, enumere los parámetros de entrada/ salida, ubicación y tipo.

b) Calcule el tiempo total de ejecución para F_{MCLK}=1MHz.

c) Ensamble manualmente el programa. ¿Cuántas palabras ocupa el código?

2.- [3 puntos] Se desea realizar la función `IncTime` que recibe una información de hora y la incrementa en un segundo:

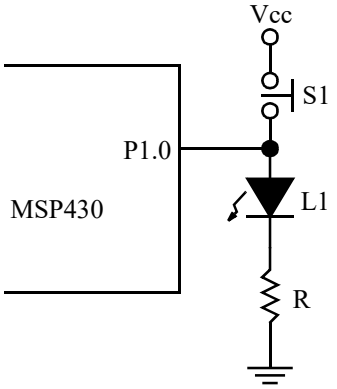
```
int IncTime (unsigned long *hms);
```



`hms` es un puntero a un entero largo sin signo que almacena, en BCD empaquetado, información temporal en formato HHMMSS (dos dígitos BCD para hora, minutos y segundos):

La función sigue el convenio de llamada de C y devuelve 1 si se produce un cambio de día (la hora de entrada es 23:59:59), -1 si la hora es incorrecta y 0 en el resto de los casos. Ignore el byte alto de la información horaria (zona rayada). La hora se guarda en formato de 24h. Escriba `IncTime` en ensamblador del MSP430. NOTA: La nueva hora se guarda en el mismo sitio que la original. Se recomienda el uso de las instrucciones que soporten aritmética BCD.

3.- [4 puntos] Observe el circuito de la figura. Inicialmente el puerto se configurará en modo entrada para poder leer el estado del pulsador. Cuando el usuario lo acciona, se enciende el led sin intervención del MSP. Se desea hacer un programa que mantenga el led encendido 1 segundo más desde la liberación del pulsador. Una vez que pase ese tiempo, se volverá al estado inicial. El programa en ensamblador debe funcionar totalmente por interrupciones, con ayuda del Timer A0 y operar en el modo de mayor bajo consumo. Considere los pulsadores libres de rebotes, el perro guardián desactivado, los puertos desbloqueados, pila inicializada, ACLK=LFXT en marcha con cristal de 32768Hz y TA0.CCR0 como función primaria del P1.0, CCI0A cuando es entrada y OUT0 cuando es salida.



SISTEMAS BASADOS EN MICROPROCESADOR **Segunda convocatoria 2023/2024**

Apellidos:.....

Nombre:.....

P1	P2	P3

Duración 4:00 horas

1.- [3 puntos] Sea el siguiente código:

```
;                               Emulación       CiclosCódigo máquina
;-------------------------------------------------------------------------------
QuienSabe   clr.w  r14          ;_____________|___|_____________________________
            mov.w  #8, r15      ;_____________|___|_____________________________
bucle       rla.b  r12          ;_____________|___|_____________________________
            adc.w  r14          ;_____________|___|_____________________________
            dec.w  r15          ;_____________|___|_____________________________
            jnz    bucle        ;_____________|___|_____________________________
            mov.w  r14, r12     ;_____________|___|_____________________________
            ret                 ;_____________|___|_____________________________
```

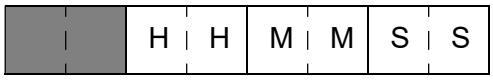
a) Analícelo en indique qué hace. Señale si sigue o no el convenio de llamada de C. En caso afirmativo, muestre el prototipo. En caso negativo, enumere los parámetros de entrada/ salida, ubicación y tipo.

b) Calcule el tiempo total de ejecución para F_{MCLK}=1MHz.

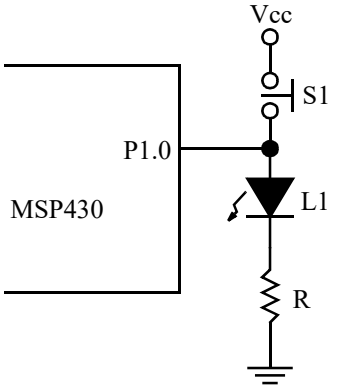
c) Ensamble manualmente el programa. ¿Cuántas palabras ocupa el código?

2.- [3 puntos] Se desea realizar la función `IncTime` que recibe una información de hora y la incrementa en un segundo:

```
int IncTime (unsigned long *hms);
```



`hms` es un puntero a un entero largo sin signo que almacena, en BCD empaquetado, información temporal en formato HHMMSS (dos dígitos BCD para hora, minutos y segundos):

La función sigue el convenio de llamada de C y devuelve 1 si se produce un cambio de día (la hora de entrada es 23:59:59), -1 si la hora es incorrecta y 0 en el resto de los casos. Ignore el byte alto de la información horaria (zona rayada). La hora se guarda en formato de 24h. Escriba `IncTime` en ensamblador del MSP430. NOTA: La nueva hora se guarda en el mismo sitio que la original. Se recomienda el uso de las instrucciones que soporten aritmética BCD.

3.- [4 puntos] Observe el circuito de la figura. Inicialmente el puerto se configurará en modo entrada para poder leer el estado del pulsador. Cuando el usuario lo acciona, se enciende el led sin intervención del MSP. Se desea hacer un programa que mantenga el led encendido 1 segundo más desde la liberación del pulsador. Una vez que pase ese tiempo, se volverá al estado inicial. El programa en ensamblador debe funcionar totalmente por interrupciones, con ayuda del Timer A0 y operar en el modo de mayor bajo consumo. Considere los pulsadores libres de rebotes, el perro guardián desactivado, los puertos desbloqueados, pila inicializada, ACLK=LFXT en marcha con cristal de 32768Hz y TA0.CCR0 como función primaria del P1.0, CCI0A cuando es entrada y OUT0 cuando es salida.