



*No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador
Se contestará sólo en el espacio reservado al efecto, pudiendo utilizar la cara posterior de la misma hoja.
No se pueden separar las hojas ni presentar hojas adicionales. Todas las respuestas deben estar justificadas.*

PROBLEMA 1 (5 puntos):

Se ha de diseñar un sistema de control y actuación en emergencias que cuenta con los siguientes dispositivos:

- Un detector de temperatura entre -15 y $+70$ grados, que proporciona a su salida una tensión entre 0 y 3V proporcional a la temperatura detectada.
- Un actuador sobre la intensidad del aire acondicionado, controlado por una tensión continua entre 0 y 3V, de tal manera que a 0V estará apagado, e irá aumentando su intensidad hasta proporcionar la máxima al recibir 3V.
- Un actuador sobre una bomba de calor, controlado por un interfaz digital de 4 bits (B3 B2 B1 B0) de tal forma que 0000 indicará que la bomba de calor está apagada, y su valor de intensidad irá incrementándose hasta que con 1111 se consigue la máxima potencia de emisión de calor.

El sistema de control y actuación debe comportarse de tal manera que si la temperatura de la sala es superior a $+22$ grados, se arranque el aire acondicionado con una intensidad proporcional a la diferencia entre la temperatura actual y dicho umbral. De la misma forma, si la temperatura actual es inferior a $+18$ grados, se arranque la bomba de calor con una intensidad proporcional a la diferencia entre ese umbral y la temperatura actual.

Para diseñar dicho sistema, utilizará un microcontrolador STM32L152RB, con todos sus pines disponibles (no sólo los disponibles en la placa STM-Discovery).

SE PIDE (sin utilizar interrupciones):

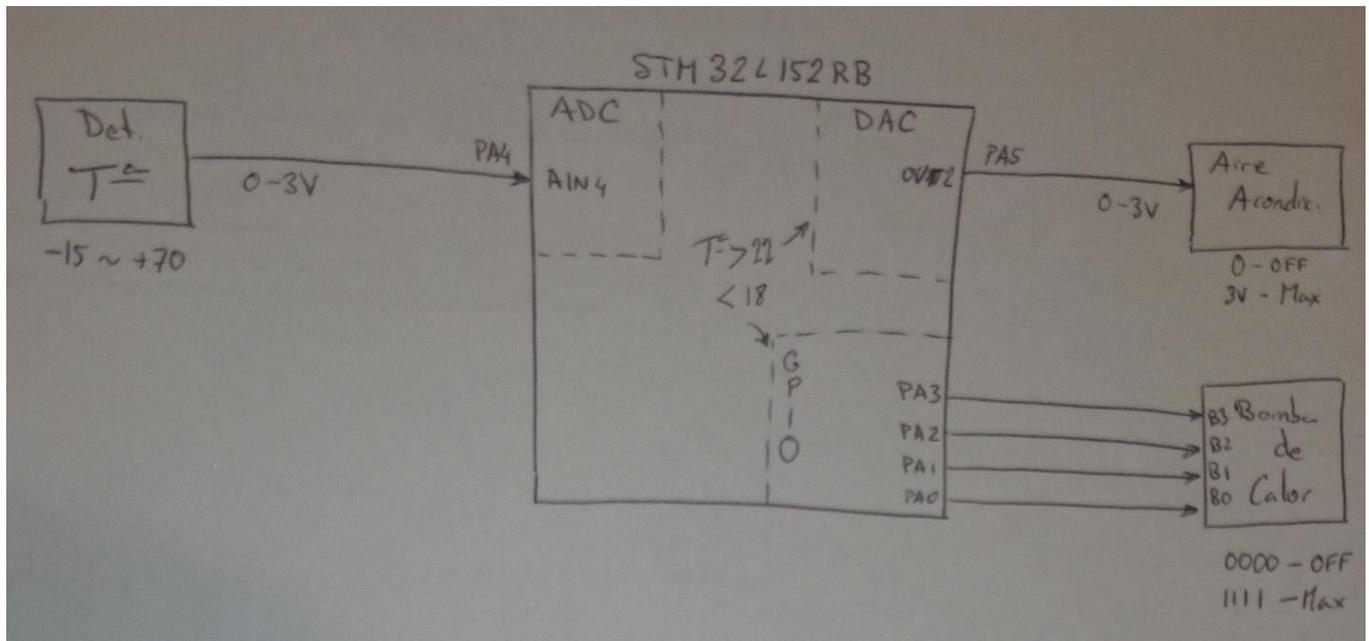
- 1) Diagrama de Bloques del sistema. (20%)
- 2) Indique qué periféricos utiliza y cómo los configuraría (justifique las decisiones tomadas). (30%)
- 3) Diagrama de flujo. (30%)
- 4) Escriba el código correspondiente a la función que convierte la temperatura actual en la intensidad a solicitar al aire acondicionado y a la bomba de calor. (20%)

SOLUCIÓN:

1) *Atendiendo al enunciado, se necesita conectar el microcontrolador al sensor de temperatura a través de un ADC, y como salidas a un actuador analógico para el Aire Acondicionado (A/A) a través de un DAC y a un interfaz digital por 4 pines del GPIO para la bomba de calor. De esta forma el diagrama de bloques queda así:*



No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador
Se contestará sólo en el espacio reservado al efecto, pudiendo utilizar la cara posterior de la misma hoja.
No se pueden separar las hojas ni presentar hojas adicionales. **Todas las respuestas deben estar justificadas.**



2) Se utilizan 3 periféricos, el GPIOA, el ADC y el DAC. No se utiliza interrupciones en ninguno por indicación del enunciado.

GPIOA:

- GPIOA->MODER:
 - Bits 11 - 10: 11, como analógico para DAC
 - Bits 9-8: 11, como analógico para ADC
 - Bits 7-0: 0101 0101, para PA3-PA0 como salidas digitales
- GPIOA->BSRRH = 0x000000FF; para poner las salidas todas a 0

ADC: Conversión por un solo canal (el canal 4), de forma continua y con 8 bits.

- ADC1->CR1:
 - Bit 26 = 0, para que no haya IRQ por overrun
 - Bits 25-24 = 10, para resolución a 8 bits, ya que no se necesita una precisión muy alta al sólo tratarse de 85 grados de diferencia y que los umbrales sean a nivel de grado.
 - Bit 8 = 0, sin Scan, porque sólo es un canal
 - Bit 5 = 0; sin IRQ por fin de conversión
- ADC1->CR2:
 - Bit 31 = 0; para tener parado el ADC, aunque luego se pondrá a 1 en el programa
 - Bit 11 = 0; para poner el resultado alineado a la derecha



*No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador
Se contestará sólo en el espacio reservado al efecto, pudiendo utilizar la cara posterior de la misma hoja.
No se pueden separar las hojas ni presentar hojas adicionales. Todas las respuestas deben estar justificadas.*

- *Bit 10 = 1; para que avise por fin de conversión con cada conversión*
- *Bits 6-4 = 000; sin retardo*
- *Bit 1 = 1; conversión continua*
- *Bit 0 = 0; ADC apagado, cuando se termine de configurar se pondrá a 1*
- *ADC1->SQR1:*
 - *Bits 24-20 = 00000, porque sólo hay un canal*
- *ADC1->SQR5:*
 - *Bits 4-0 = 00100, canal 4*

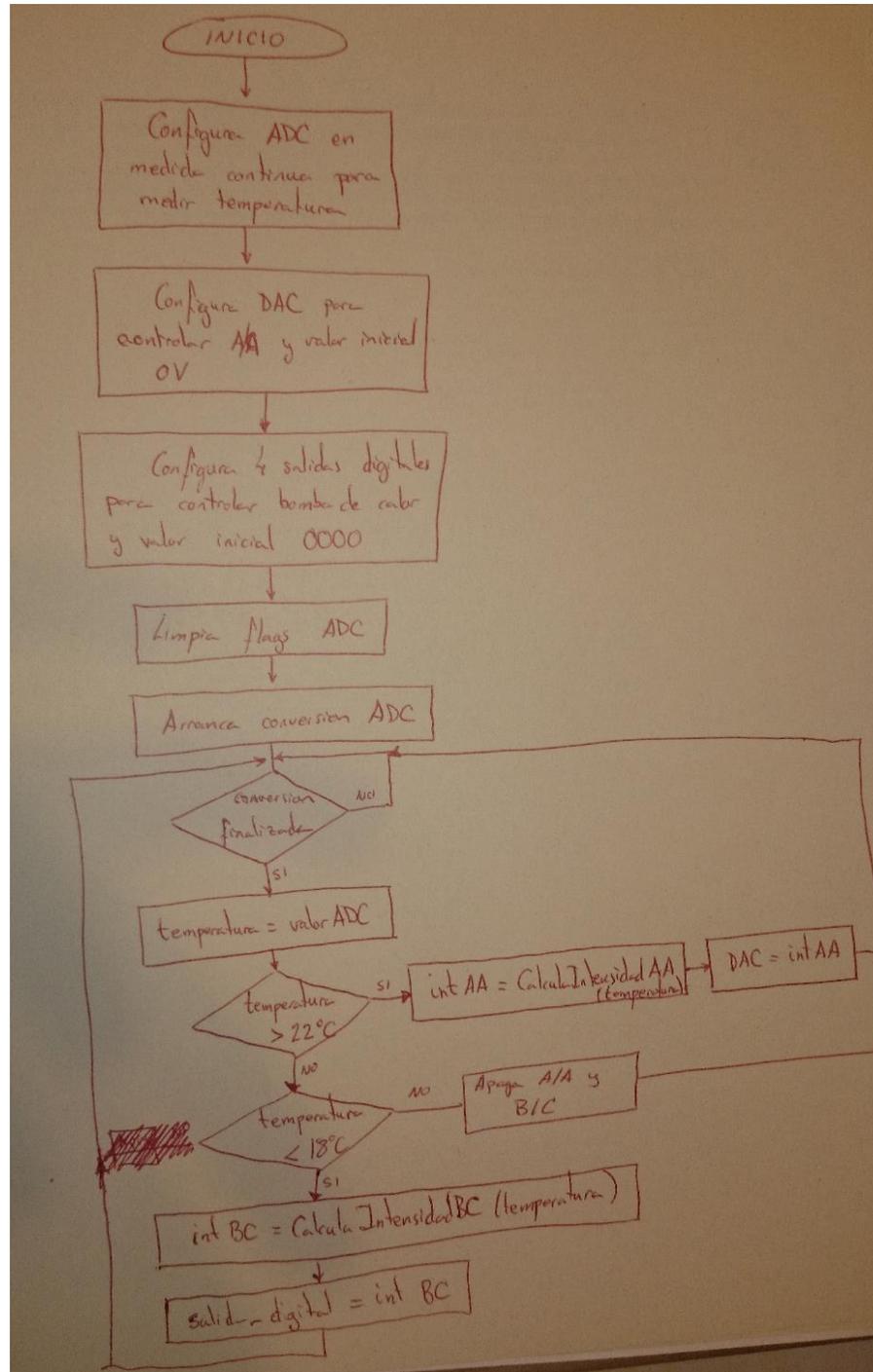
DAC: Conversión a 8 bits (si el sensor es a 8 bits, no se entendería tener una resolución mayor que la del ADC), por canal 2, que es el que está conectado al PA5

- *DAC->CR:*
 - *Bits 17-16 = 01; habilitado el buffer del canal 2 y habilitado el canal 2*
- *El dato se escribirá en DAC->DHR8R2*

3) *El diagrama de flujo sería de la siguiente forma:*

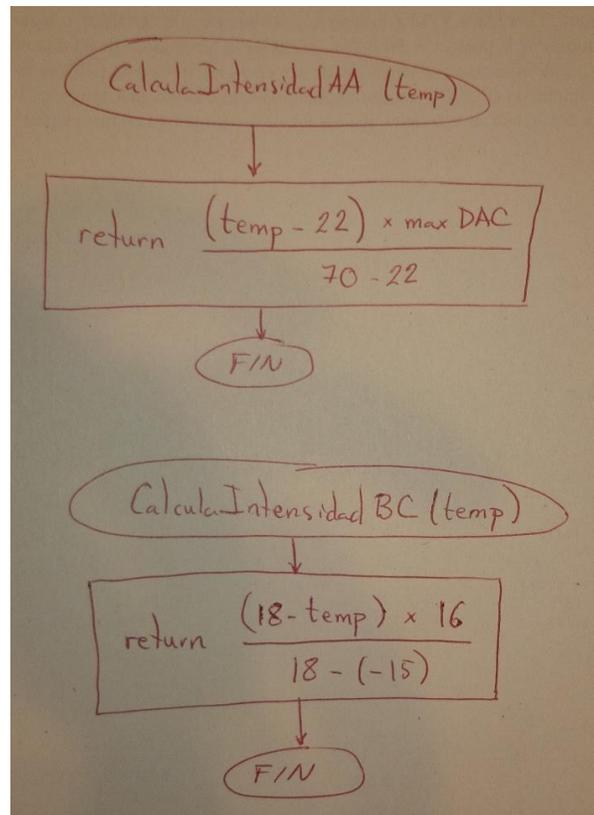


No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador
Se contestará sólo en el espacio reservado al efecto, pudiendo utilizar la cara posterior de la misma hoja.
No se pueden separar las hojas ni presentar hojas adicionales. **Todas las respuestas deben estar justificadas.**





No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador
Se contestará sólo en el espacio reservado al efecto, pudiendo utilizar la cara posterior de la misma hoja.
No se pueden separar las hojas ni presentar hojas adicionales. **Todas las respuestas deben estar justificadas.**



4) Tal y como se ha configurado el ADC a 8 bits, podemos hacer la traducción de temperaturas a valores del ADC, y utilizar esos valores en el programa:

- -15°C es 0
- $+70^{\circ}\text{C}$ es 255
- $+18^{\circ}\text{C}$ es $(18 - (-15)) * 255 / (70 - (-15)) = 99$
- $+22^{\circ}\text{C}$ es $(22 - (-15)) * 255 / (70 - (-15)) = 111$

Por lo tanto el código empezaría definiendo fuera del código las constantes de las temperaturas:

```
#define TEMP18 99  
#define TEMP22 111
```

Se necesitará tener definidas dos variables locales, que nunca tendrán valor superior a 255, por lo que pueden ser unsigned char:



*No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador
Se contestará sólo en el espacio reservado al efecto, pudiendo utilizar la cara posterior de la misma hoja.
No se pueden separar las hojas ni presentar hojas adicionales. **Todas las respuestas deben estar justificadas.***

*unsigned char temperatura;
unsigned char aux;*

Y el código sería:

```
temperatura = ADC1->DR  
if (temperatura > TEMP22)  
    DAC1->DHR8R2 = (unsigned char)(((temperatura - TEMP22) * 255)/(255 - TEMP22));  
else if (temperatura < TEMP18) {  
    aux = ((TEMP18 - temperatura)*16)/TEMP18;  
    GPIOA->BSRRH = 0x0F;  
    GPIOA->BSRRL = (aux & ( 0x0F ));  
}  
else {  
    DAC->DHR8R2 = 0;  
    GPIOA->BSRRH = 0x0F;  
}
```



*No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador
Se contestará sólo en el espacio reservado al efecto, pudiendo utilizar la cara posterior de la misma hoja.
No se pueden separar las hojas ni presentar hojas adicionales. Todas las respuestas deben estar justificadas.*

PROBLEMA 2 (5 puntos):

El código siguiente se ha de ejecutar en la placa STM-Discovery, y su intención es arrancar o parar una conversión analógica continua, a través de la pulsación del botón USER de la placa (si se pulsa una vez, se arranca, si se vuelve a pulsar, se para), mostrando el resultado de las conversiones en el LCD si está en modo conversión, y en otro caso, limpiando el LCD y encendiendo el led azul cuando esté apagado. Dicho código está lleno de errores, por lo que se le solicite que los localice y proponga cómo arreglarlos.

SOLUCIÓN:

Los errores localizados son los siguientes:

- *Línea 5: estado_ant está definida como variable global, aunque es sólo local al main. Habría que mover esa línea a una nueva entre las líneas 14 y 15.*
- *Línea 7: El nombre de la RAI es incorrecto, ya que todo el programa está realizado para que salte la interrupción por el botón USER, que está conectado al PA0, y por tanto la EXTI correspondiente debería ser la EXTIO. Por lo tanto la línea 7 debería ser:*
 - `void EXTIO_IRQHandler(void) {`
- *Entre la línea 11 y 12: En la RAI no se limpia nunca el flag, por lo que la interrupción seguirá saltando todo el tiempo. Entre esas dos líneas habría que añadir:*
 - `EXTI->PR = 0x01;`
- *Líneas 18 y 19: las operaciones para la configuración del modo de funcionamiento del PB6 están mal por dos motivos. Primero porque debería ser en los dos bit 01, y para eso habría que hacer un AND en el bit más significativo y un OR en el bit menos significativo. Y segundo, porque tal y como están las máscaras, peciamente se manipulan todos los bits menos los que realmente se quieren manipular. En su lugar esas dos líneas deberían ser:*
 - `GPIOB->MODER &= ~(1 << (6*2 +1));`
 - `GPIOB->MODER |= (1 << (6*2));`
- *Líneas 23 y 24: el modo de operación de los pines del puerto A está mal, ya que se están manipulando todos los bits, salvo sólo los bits 0 y 1, que ambos deben estar a cero. Esas líneas no deberían hacer asignación, sino máscara de AND para poner a cero dichos bit. Por lo tanto esas líneas deberían ser:*
 - `GPIOA->MODER &= ~(1 << (0*2 +1));`
 - `GPIOA->MODER &= ~(1 << (0*2));`



*No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador
Se contestará sólo en el espacio reservado al efecto, pudiendo utilizar la cara posterior de la misma hoja.*

No se pueden separar las hojas ni presentar hojas adicionales. Todas las respuestas deben estar justificadas.

- *Línea 26: supuestamente esta línea debería poner un pin del puerto A, por ejemplo el PA4, a modo analógico (MODER a 11), y sin embargo el PA4 lo deja a 0 entrada digital y el 5 lo deja a salida digital. La línea debería ser, (teniendo en cuenta que el canal que se utiliza en el ADC es el canal 4):*
 - *GPIOA->MODER |= 0x00000300;*
- *Líneas 46 y 49: la funcionalidad del led azul está al revés, se enciende cuando debería estar apagado y se apaga cuando debería encenderse. Esto se solucionaría haciendo que, respectivamente esas dos líneas fuesen:*
 - *GPIOB->BSRRL = (1<<6);*
 - *default: GPIOB->BSRRH = (1<<6);*
- *Línea 47: la función LCD_Limpia no tiene parámetros de entrada, por lo que está definida con parámetro (void). Sin embargo, ha de llamarse de la siguiente manera:*
 - *LCD_Limpia();*
- *Línea 53: la función Bin2Ascii no está definida, a no ser que estuviese dentro de Biblioteca_SDM.h. Como no tenemos seguridad, no se considera un error, pero sería cuestión de comprobarlo.*
- *Línea 53: ni valor ni texto están definidos. Deberían definirse como variables locales al método main. Por ejemplo, entre las líneas 14 y 15:*
 - *short valor;*
 - *unsigned char texto[6];*
- *La funcionalidad no está conseguida, porque cuando se da el caso de que se hacen las conversiones, sólo se muestra la primera de las conversiones. Para arreglar eso, habría que poner dentro del while(1) una condición que dijese que si el estado es el de conversiones, en ese caso se verificase el valor del flag, se limpiase y se mostrase el resultado, es decir, entre las líneas 57 y 58 debería haber:*
 - *if (estado == 1) {*
 - *Mover aquí la línea 50*
 - *Mover aquí la línea 52*
 - *ADC1->SR &= 0x0040;*
 - *Mover aquí las líneas 53 y 54*
 - *}*
- *Entre las líneas 47 y 48, hay que parar el ADC, insertando:*
 - *ADC1->CR2 &= ~(0x40000000);*



UNIVERSIDAD CARLOS III DE MADRID
Sist. Dig. Basados en Microprocesador

2 de marzo de 2015

(Dpto. de Tecnología Electrónica)
(Gr. Ing. Telemática)

1er PARCIAL (100 minutos)

No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador

Se contestará sólo en el espacio reservado al efecto, pudiendo utilizar la cara posterior de la misma hoja.

*No se pueden separar las hojas ni presentar hojas adicionales. **Todas las respuestas deben estar justificadas.***



*No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador
Se contestará sólo en el espacio reservado al efecto, pudiendo utilizar la cara posterior de la misma hoja.
No se pueden separar las hojas ni presentar hojas adicionales. Todas las respuestas deben estar justificadas.*

APLICACIÓN

```
1 #include "stm32l1xx.h"
2 #include "Biblioteca_SDM.h"
3
4 unsigned char estado = 0;
5 unsigned char estado_ant = 0;
6
7 void EXTI1_IRQHandler(void) {
8     if (EXTI->PR!=0) {
9         estado++;
10        if (estado > 1) estado = 0;
11    }
12 }
13
14 int main(void){
15     Init_SDM();
16     Init_LCD();
17
18     GPIOB->MODER |= ~(1 << (6*2 +1));
19     GPIOB->MODER &= (1 << (6*2));
20     GPIOB->OTYPER &= ~(1 << 6);
21     GPIOB->OSPEEDR &= ~(1 << (6*2 +1));
22     GPIOB->OSPEEDR &= ~(1 << (6*2));
23     GPIOA->MODER = ~(1 << (0*2 +1));
24     GPIOA->MODER = ~(1 << (0*2));
25     GPIOA->PUPDR &= ~(11 << (0*2));
26     GPIOA->MODER |= 0x00000400;
27     ADC1->CR2 &= ~(0x00000001);
28     ADC1->CR1 = 0x00000002;
29     ADC1->CR2 = 0x00000412;
30     ADC1->SMPR1 = 0;
31     ADC1->SMPR2 = 0;
32     ADC1->SMPR3 = 0;
33     ADC1->SQR1 = 0x00000000;
34     ADC1->SQR5 = 0x00000004;
35     ADC1->CR2 |= 0x00000001;
36     EXTI->FTSR &= 0x01;
37     EXTI->RTSR &= ~(0x01);
38     SYSCFG->EXTICR[0] = 0;
39     EXTI->IMR |= 0x01;
40     NVIC->ISER[0] |= (1 << 6);
41
42     while (1) {
43         if (estado_ant != estado) {
44             estado_ant = estado;
45             switch(estado) {
46                 case 0: GPIOB->BSRRH = (1<<6);
47                        LCD_Limpia(void);
48                        break;
49                 default: GPIOB->BSRRL = (1<<6);
50                        while ((ADC1->SR&0x0040)==0);
51                        ADC1->CR2 |= 0x40000000;
52                        valor = ADC1->DR;
53                        Bin2Ascii(valor,&texto[0]);
54                        LCD_Texto(texto);
55                        break;
56            }

```



UNIVERSIDAD CARLOS III DE MADRID
Sist. Dig. Basados en Microprocesador

2 de marzo de 2015

(Dpto. de Tecnología Electrónica)
(Gr. Ing. Telemática)

1er PARCIAL (100 minutos)

No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador

Se contestará sólo en el espacio reservado al efecto, pudiendo utilizar la cara posterior de la misma hoja.

*No se pueden separar las hojas ni presentar hojas adicionales. **Todas las respuestas deben estar justificadas.***

57 }
58 }
59 }



UNIVERSIDAD CARLOS III DE MADRID
Sist. Dig. Basados en Microprocesador

2 de marzo de 2015

(Dpto. de Tecnología Electrónica)
(Gr. Ing. Telemática)

1er PARCIAL (100 minutos)

No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador

Se contestará sólo en el espacio reservado al efecto, pudiendo utilizar la cara posterior de la misma hoja.

*No se pueden separar las hojas ni presentar hojas adicionales. **Todas las respuestas deben estar justificadas.***