

Problema 1 (60 minutos – 4 puntos)

Se ha desarrollado un sistema de cerradura electrónica para la puerta de acceso a uno de los laboratorios de la universidad. Para esta aplicación se ha utilizado el microcontrolador de referencia del curso funcionando con una frecuencia de oscilación de 32MHz y alimentado entre 0 y 3V.

Como interfaz de usuario el sistema dispone de:

- Un dispositivo que se comunica con el microcontrolador mediante una comunicación serie asíncrona (DISPOSITIVO_SERIE). Este dispositivo está formado por una pantalla en la que pueden visualizarse los mensajes enviados desde el microcontrolador y un teclado con el que pueden introducirse los números de la clave de acceso. Esta clave de acceso consta de 4 dígitos.
- Un botón de cancelación (BOTON_CANCEL) que permite cancelar la introducción del código de acceso en cualquier momento. Si se pulsa este botón, es necesario volver a introducir la clave completa de nuevo.

Además, la puerta de acceso se encuentra conectada mediante un dispositivo intermedio (DISPOSITIVO_APERTURA) a un pin de salida digital del microcontrolador. Cuando este pin de salida tiene un “1” lógico la puerta se abre (con un “0” lógico en este pin de salida, la puerta se cierra).

El microcontrolador también tiene conectado a un pin de salida digital un dispositivo de seguridad de bloqueo de la puerta (DISPOSITIVO_BLOQUEO). Cuando el pin del microcontrolador al que se encuentra conectado el dispositivo de bloqueo se pone a un “1” lógico el dispositivo actúa y la puerta se bloquea. El bloqueo de la puerta se produce después de 3 intentos erróneos de introducir la clave de acceso. **El desbloqueo de la puerta se hace externamente al sistema que se presenta**, volviendo a poner en este pin de salida un “0” lógico.

Por último, se dispone de un ALTAVOZ conectado al microcontrolador que emite una alarma sonora para avisar que la puerta se encuentra bloqueada.

El esquema del sistema completo es el que se muestra en la figura 1:

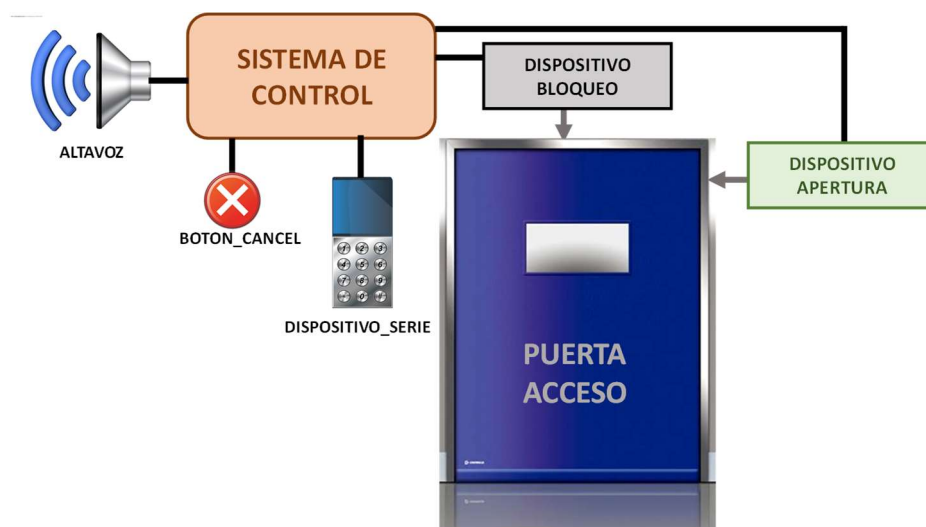


Figura 1. Esquema del sistema de acceso con cerradura electrónica

El código desarrollado para la aplicación es el que se muestra al final de este enunciado (se recomienda ver y analizar el código antes de empezar a contestar a las preguntas que se plantean):

Se pide:

1. Represente el esquema del hardware (diagrama de bloques) para esta aplicación. Dibuje el microcontrolador como un bloque, indicando claramente todos los recursos del microcontrolador utilizados y qué pines del microcontrolador se conectan a cada una de las entradas o salidas de todos los elementos que componen el sistema (cada elemento debe estar claramente identificado en el diagrama con el mismo nombre con que se denota en la figura 1). Además, rellene la siguiente tabla (añada todas las filas necesarias). **(25%)**

PIN MICRO	HW CONECTADO	TIPO (E/S, DIGITAL/ ANALOGICO)	JUSTIFICACIÓN

2. Represente el diagrama de flujo del programa principal, función/es y rutina/s de atención a interrupción del código. **(30%)**

3. ¿Con qué recurso del microcontrolador se controla el tiempo que la puerta de acceso permanece abierta, y cuánto tiempo permanece la puerta abierta después de introducir el código de entrada? Justifique su respuesta incluyendo todos los cálculos necesarios para obtenerla. **(15%)**

4. Indique la configuración completa de la comunicación serie utilizada para la introducción del código de acceso. Indique la/s modificación/es necesarias en el código para que los parámetros de esta comunicación serie pasen a ser 57600, 8, N, 1. **(15%)**

5. Represente dos períodos completos de la forma de onda de la señal que se aplica al altavoz (acotándola claramente en amplitud y tiempo). Indique la/s modificación/es necesarias en el código que esta señal sea una señal entre 0 y 3V con una frecuencia de 1Hz **(15%)**.

CÓDIGO PROGRAMA CERRADURA ELECTRÓNICA

```
1 #include "stm32l1xx.h"
2
3
4 void EscribeMensaje(unsigned char* mensaje, unsigned char longitud_mensaje);
5
6 unsigned char nueva_entrada_codigo=1;
7 unsigned char posicion_caracter_recibido=0;
8 unsigned char codigo_finalizado=0;
9 unsigned char codigo_apertura[4]={'1','2','3','4'};
10 unsigned char codigo_introducido[4]={0,0,0,0};
11
12 void EXTI0_IRQHandler(void) {
13     if (EXTI->PR!=0){
14         nueva_entrada_codigo=1;
15         EXTI->PR =0x00001;
16     }
17 }
18 void TIM2_IRQHandler(void) {
19     if (TIM2->SR&0x0002!=0){
20         TIM2->CCR1 += 1000;
21         TIM2->SR = 0x0000;
22     }
23 }
24
25 void USART1_IRQHandler(void) {
26     if ((USART1->SR&0x0020)!=0){
27         codigo_introducido[posicion_caracter_recibido]=USART1->DR;
28         if (posicion_caracter_recibido==3){
29             codigo_finalizado=1;
30             posicion_caracter_recibido=0;
31         }
32         else posicion_caracter_recibido++;
33     }
34 }
35
36 void Inicializa(void){
37     GPIOB->MODER &= ~(0x0000F000);
38     GPIOB->MODER |= (0x0000A000);
39     GPIOB->AFR[0] &= ~(0xFF000000);
40     GPIOB->AFR[0] |= 0x77000000;
41     USART1->CR1 = 0x0000002C;
42     USART1->CR2 = 0x00000000;
43     USART1->BRR = 0x00000115;
44     USART1->CR1 |= 0x2000;
45
46     GPIOA->MODER &= ~(1 << (0*2 +1));
47     GPIOA->MODER &= ~(1 << (0*2));
48     SYSCFG->EXTICR[0]&= ~(0x000F);
49     EXTI->RTSR |= (0x00000001);
50     EXTI->FTSR &= ~ (0x00000001);
51     EXTI->IMR |= 0x00000001;
52
53     GPIOA->MODER &= ~(1 << (12*2 +1));
54     GPIOA->MODER |= (1 << (12*2));
55     GPIOA->BSRR= (1<<12)<<16;
56
57     GPIOA->MODER &= ~(1 << (11*2 +1));
58     GPIOA->MODER |= (1 << (11*2));
59     GPIOA->BSRR= (1<<11)<<16;
60
61     GPIOA->MODER |= (1 << (5*2+1));
62     GPIOA->MODER &= ~(1 << (5*2));
63     GPIOA->AFR[0]&= ~(0x00F00000);
64     GPIOA->AFR[0]|=(0x00100000);
65     TIM2->CR1 = 0x0000;
66     TIM2->CR2 = 0x0000;
67     TIM2->SMCR = 0x0000;
68     TIM2->PSC = 32;
69     TIM2->CNT = 0;
70     TIM2->ARR = 0xFFFF;
71     TIM2->CCR1= 1000;
72     TIM2->CCMR1 = 0x0030;
73     TIM2->CCER = 0x0001;
74     TIM2->EGR |= 0x0001;
75     TIM2->SR = 0;
76     TIM2->DIER=0x0002;
77 }
```

```

78     TIM4->CR1 = 0x0000;
79     TIM4->CR2 = 0x0000;
80     TIM4->SMCR = 0x0000;
81     TIM4->PSC = 32000;
82     TIM4->ARR = 0xFFFF;
83     TIM4->CCR1= 10000;
84     TIM4->CCMR1 = 0x0000;
85     TIM4->CCER = 0x0000;
86     TIM4->EGR |= 0x0001;
87     TIM4->SR = 0;
88     TIM4->DIER=0x0000;
89
90     EXTI->PR=0x0001;
91     NVIC->ISER[0] |= (1 << 6);
92     NVIC->ISER[0] |= (1 << 28);
93     NVIC->ISER[1] |= (1 << (37-32));
94
95 }
96
97 void EscribeMensaje(unsigned char* mensaje, unsigned char longitud_mensaje){
98     unsigned char i=0;
99     for (i=0;i<longitud_mensaje;i++){
100         while ((USART1->SR & 0x0080)== 0);
101         USART1->DR = mensaje[i];
102     }
103 }
104
105 int main(void){
106     unsigned char intentos=0;
107     unsigned char mensaje_puerta_bloqueada[]="\nPUERTA BLOQUEADA\n";
108     unsigned char mensaje_puerta_abierta[]="\nPUERTA ABIERTA\n";
109     unsigned char mensaje_introducir_codigo[]="\nINTRODUZCA CODIGO\n";
110     unsigned char mensaje_codigo_incorrecto[]="\nCODIGO INCORRECTO\n";
111
112     Inicializa();
113     while (1) {
114         if ((GPIOA->IDR&0x1000)!=0){
115             EscribeMensaje(mensaje_puerta_bloqueada, sizeof(mensaje_puerta_bloqueada));
116             TIM2->CR1 |= 0x0001;
117             while ((GPIOA->IDR&0x1000)!=0); //ESPERA DESBLOQUEO EXTERNO PUERTA
118             GPIOA->BSRR=(1<<12)<<16;
119             TIM2->CR1 &= ~(0x0001);
120             intentos=0;
121         }
122         if ((GPIOA->IDR&0x0800)!=0){
123             EscribeMensaje(mensaje_puerta_abierta,sizeof(mensaje_puerta_abierta));
124             TIM4->CNT=0;
125             TIM4->CR1 |= 0x0001;
126             while ((TIM4->SR&0x0002)==0);
127             TIM4->SR = 0;
128             TIM4->CR1 &= ~(0x0001);
129             GPIOA->BSRR=(1<<11)<<16;
130             intentos=0;
131         }
132         if(nueva_entrada_codigo==1){
133             EscribeMensaje(mensaje_introducir_codigo,sizeof(mensaje_introducir_codigo));
134             nueva_entrada_codigo=0;
135             posicion_caracter_recibido=0;
136         }
137         if (codigo_finalizado==1){
138             codigo_finalizado=0;
139             if(codigo_introducido[0]==codigo_apertura[0]&codigo_introducido[1]==codigo_apertura[1]&codigo_introducido[2]==codigo_apertura[2]&
140             codigo_introducido[3]==codigo_apertura[3]) GPIOA->BSRR= (1<<11);
141             else{
142                 EscribeMensaje(mensaje_codigo_incorrecto,sizeof(mensaje_codigo_incorrecto));
143                 intentos++;
144                 if (intentos==3) GPIOA->BSRR= (1<<12);
145             }
146             nueva_entrada_codigo=1;
147         }
148     }
149 }

```

Problema 2 (90 minutos – 6 puntos)

Basándose en un microprocesador STM32L152RB con pclk de 32MHz (no conectado a la placa de desarrollo utilizada en el curso), se tiene que diseñar un sistema para controlar la temperatura y la humedad relativa del interior del habitáculo de un coche de gama media que cumpla con los siguientes **requisitos**:

- El microcontrolador debe tener conectado 2 sensores de temperatura que suministran una tensión entre 0 y 3,3V para indicar la temperatura, la cual variará, como más rápido, cada 5 segundos, siendo el valor máximo de temperatura de 60°C y el valor mínimo de -10°C. Los 2 sensores sirven para tomar la medida de temperatura en la parte delantera y trasera del habitáculo del vehículo, de manera que el programa debe calcular la media de los 2 sensores para tomar esa temperatura como temperatura real del habitáculo y a partir de ahí hacer lo siguiente (**se supone que el programa de control va a tener de alguna manera el valor de consigna deseado por el usuario**, no hay que preocuparse como lo consigue):
 - Si la temperatura media se aleja menos de 0,1 grado de la deseada por el conductor, el microcontrolador desconecta el sistema auxiliar que regula la temperatura del habitáculo.
 - Si la diferencia entre la temperatura media y la temperatura deseada es superior a 0,1 grados (tanto positiva, como negativa), el microcontrolador activa el sistema auxiliar que regula la temperatura del habitáculo para conseguir la temperatura deseada (enfriando o calentando en función de cada caso), el cual es controlado por una señal PWM de 100Hz proveniente del microcontrolador, de manera que el DC empieza al 10% y va subiendo de 10% en 10% hasta el 90% con cada 0,1° que se aleje de la temperatura deseada, es decir, si se aleja 0,1°, el DC es 10%, si se aleja 0,2°, el DC es 20% y así hasta un máximo del 90% en el DC.
- El microcontrolador debe tener además conectado un sensor de humedad relativa en el centro del habitáculo del vehículo que cada minuto emite una señal eléctrica entre 3,3 y 0V, siendo el valor máximo una humedad relativa del 100°C y el valor mínimo de 0°C y hacer lo siguiente:
 - Si la humedad relativa está entre el 40% y el 70%, el microcontrolador desconecta el sistema auxiliar de compensación de humedad relativa.
 - Si la humedad relativa está por debajo del 40% o por encima del 70%, el sistema genera una señal digital que activa el sistema auxiliar de compensación de humedad relativa ('0' – sistema de compensación apagado; '1' – sistema de compensación encendido).
- El sistema, además, recibe una señal de entrada digital externa que le indica, mediante flancos, que la plataforma donde está instalado el sistema se encuentra operativa (es decir, una especie de "señal de vida"). Dicha señal debería tener siempre menos de 10 segundos entre flancos para indicar que el sistema está operativo. Es necesario medir el tiempo que transcurre entre los flancos de esta señal, de forma que:
 - Si es menor de 10 segundos no se hace nada, pero si es mayor de 10 segundos, el sistema enciende un LED para indicar que algo raro ocurre y deshabilita el sistema de control de temperatura y humedad. Además, en ese caso, desactivará todas las señales de salida.

- El usuario dispone de un pulsador para reinicializar la funcionalidad descrita anteriormente en cualquier momento, desactivando además todas las señales de salida del sistema. (Si pulsador es '1' = Sistema en funcionamiento normal; si es '0' = Reinicializa la funcionalidad) y de un interruptor para encenderlo o apagarlo ('0' = Apagado / '1' = Encendido).

Necesidades especiales de diseño:

- Por necesidades de optimización, todo se debe realizar con el puerto A del micro excepto la señal para el LED que debe ser del puerto B.
- Por necesidades de diseño, los 4 canales de cada temporizador se deben utilizar para los mismos modos del mismo, no se pueden mezclar modos en los canales. Y además no se puede utilizar el TIM1.
- La señal de vida se debe evaluar obligatoriamente utilizando alguno de los modos de funcionamiento de los temporizadores que no sea el modo TOC, ni tampoco se puede hacer con señales digitales normales y espera activa en programa.

Con estas especificaciones, optimizando los recursos y de forma razonada:

- a) Dibuje el diagrama de bloques del sistema. (20%)
- b) Indique si utilizará interrupciones, y en caso afirmativo, cuáles y para qué. (20%)
- c) Indique qué periféricos utiliza y cómo los configuraría. No escriba el código, sino el valor que tendría que tener cada uno de los bits afectados de cada uno de los registros del periférico y por qué. (40%)
- d) Dibuje el diagrama de flujo del programa de control y las posibles interrupciones. (20%)