

Enunciado

Se necesita disponer de un sistema de acceso personal mediante tarjetas.

Se dispone de aprox 50 tarjetas codificadas y un lector de códigos que emite por línea serie 19200,8,n,1 el código de la tarjeta (un byte)

Luego de RESET, el sistema estará en la etapa de programación, hasta que se pasen por el lector 10 tarjetas diferentes, mediante interrupción serie guardará los códigos en una tabla de RAM interna y después de recibir el último enciende un LED_VERDE.

A partir de ese momento, cada vez que se pasa una tarjeta por el lector el sistema controla si está en la lista grabada durante programación y si la encuentra realiza un pulso (no importa su duración) para abrir la puerta de ingreso.

Si no la encuentra realiza un pulso (no importa su duración) en LED_ROJO

Solucion

```
$bitdef.h ;manejo de a bits
```

```
$ioAT89C52.h ;incluye los registros del microcontrolador
```

```
NAME main
```

```
PUBLIC main
```

```
PUBLIC Isr_Serie
```

```
PUBLIC Interrupciones ;Declaro todas mis subrutinas
```

```
PUBLIC Recibo
```

```
PUBLIC Busqueda
```

```
;Constantes
```

```
INT_PRIORIDAD EQU 0x00 ;prioridad baja, viene en 0 pero aseguro.
```

```
IE_HABILITADAS EQU b'10010000 ;habilitación llave ES y EA
```

```
;Para TIMERS
```

```
TMOD_INICIAL EQU b'00100000 ;Pongo modo 2 el timer 1 y tmr0 no se usa
```

```
TCON_INICIAL EQU b'01000000 ;Registro de control de timers, tmr1 ON y tmr0 no se usa
```

```
RECARGA_TMR1 EQU 0xFD ;Cuenta de Recarga del Timer 1 para 19200 baudios (SMOD=1)
```

```
SCON_INICIAL EQU b'01010000 ;Modo 1 (SM0=0;SM1=1) y Recepción habilitada
```

;Puerto I/O

RX_SERIE_I EQU P3.0 ;entrada de recepción serie

LED_VERDE EQU P3.2 ;Led verde

LED_ROJO EQU P3.5 ;Led Rojo

CERRADURA EQU P3.4 ;Puerta

ASEG

org 0000h

ljmp main

ASEG

org 0023H ;vector de Interrupción Serie

ljmp Isr_Serie ;salto a Isr_Serie.

dato_rx ds 1 ;Guardo aca lo que llega por el puerto serie

hay_dato_rx ds 1 ;flag de que hay un dato recibido puerto serie. En '1', hay dato, en '0', no hay dato . RI

bandera ds 1

ORG 0030h ;punto de inicio del Programa

tabla ds 10 ;Creo la tabla para las 10 tarjetas

;-----

RSEG RCODE

main

;**Inicializaciones

call Interrupciones ;Subrutina para Inicializar Interrupciones

setb RX_SERIE_I ;aseguro puerto de recepción serie como entrada

;Inicializo variables

mov hay_dato_rx,#0x00 ;Inicializo en 0, no me lleo nada todavía!

mov R0,#tabla ;Guardo posición de la tabla en R0

mov R2,#10D ;para contar las 10 tarjetas

mov bandera, #00D ;Bandera inicia en 0

;Inicia el loop principal

Loop_ppal:

call Recibo ;subrutina para ver si lleo algo

mov A, bandera ;Bandera para saber si pase a la etapa 2

cjne A,#1, Loop_ppal ;Si paso de aca es que estoy en la etapa 2

lectura:

mov R0,#tabla ;Posiciono el puntero en el inicio de la tabla

call Busqueda ;Salto a la subrutina de busqueda

jmp lectura ;Me quedo en un loop procesando las tarjetas leidas

;fin del loop principal

;-----

;Subrutina para la recepcion de las tarjetas

Recibo:

push PSW

mov A,hay_dato_rx ;Muevo a A la bandera que se activa en la interrupcion

mov hay_dato_rx,#0x00 ;limpio el flag

cjne A,#1,Salir ;Tengo algo sigo, sino salgo

mov @R0, dato_rx ;Guardo algo en la tabla

INC R0 ;Aumento el puntero

djnz R2, Salir ;Pregunto si el contador de las 10 lleo a cero.

mov bandera, #1 ;activo bandera, quiere decir que termino de lleo las 10 tarjetas

setb LED_VERDE ;Dejo prendido el led verde

Salir:

pop PSW

ret

;-----

;Subrutina para la búsqueda de las tarjetas

Busqueda:

```
push    PSW
mov     R3,#10D           ;Para ver en las 10 posiciones de la tabla
mov     A,hay_dato_rx     ;Muevo la bandera a A
mov     hay_dato_rx,#0x00 ;limpio el flag
cjne    A,#1,Salir_buscar ;Tengo algo sigo, sino salgo
```

compara:

```
mov     A,dato_rx         ;Muevo el dato leído a A
mov     B,@R0             ;Muevo el dato de la tabla a B
cjne    A,B,saltear       ;Comparo dato recibido por serie y dato ya guardado en la tabla
setb    CERRADURA        ;Pulso para abrir puerta
clr     CERRADURA
jmp     Salir_buscar      ;Salto a Salir_buscar
```

saltear:

```
inc     R0                ;Incremento el puntero
djnz    R3, compara       ;Si no busque 10 veces vuelvo a comparar
                        ;Si ya busque en mis 10 tarjetas y no se encontro
setb    LED_ROJO          ;pulso al led rojo
clr     LED_ROJO
```

Salir_buscar:

```
pop     PSW
ret
```

;-----

Interrupciones:

```
;Cargo las constantes que defini arriba
mov  TL1, #RECARGA_TMR1      ;Recargo la parte baja del Timer 1
mov   TH1, #RECARGA_TMR1 ;Recargo la parte alta del Timer 1
mov  TMOD, #TMOD_INICIAL    ;Inicializo Timer/Counter Mode Register
mov  TCON, #TCON_INICIAL    ;Inicializo Timer/Counter Control Register
mov  SCON, #SCON_INICIAL     ;Inicializo puerto serie
mov  IP, #INT_PRIORIDAD      ;Prioridad baja para todos
mov  IE, #IE_HABILITADAS     ;Habilito las interrupciones según la cte
setb PCON                    ;Porque tengo que transmitir a 19200 baudios
ret
```

;-----

Isr_Serie:

```
push PSW                ;Resguardo PSW
```

Que_fue:

```
jnb SCON_RI, Salir_serie ;Pregunto por precaucion, pero no transmito nada.
```

Es_recepcion: ;Entro aca cuando recibo

```
cpl SCON_RI            ;Vuelvo RI a 0
mov dato_rx, SBUF       ;Guardo lo recibido en dato
mov hay_dato_rx, #01H   ;Levanto bandera de que llego algo
jmp Salir_serie         ;salto y salgo
```

Salir_serie:

```
pop PSW                ;Levanto el PSW resguardado
reti
```

;=====

END main