

## Ejercicio 13: TIMER con minutos y segundos

```
TIMER EQU 10H
PIC EQU 20H
EOI EQU 20H
N_CLK EQU 10
```

```
ORG 40
IP_CLK DW RCLK
```

```
ORG 1000H
MIN DB "00:"
SEG DB 30H
DB 30H
FIN DB ?
```

```
ORG 3000H
RCLK: PUSH AX
INC SEG ; +1 seg
CMP SEG, 36H ; 00:60
JNZ RESET
MOV SEG, 30H ; XX:00
INC MIN+1 ; +1 min
CMP MIN+1, 3AH ; 10 min
JNZ RESET
MOV MIN+1, 30H ; X0:00
INC MIN ; +10 min
CMP MIN, 36H ; 60:00
JNZ RESET
MOV MIN, 30H ; 00:00
```

```
RESET: INT 7
MOV AL, 0
OUT TIMER, AL
MOV AL, EOI
OUT PIC, AL
POP AX
IRET
```

```
ORG 2000H
CLI
MOV AL, 0FDH ; Permitidos
OUT PIC+1, AL ; IMR
MOV AL, N_CLK ; Número
OUT PIC+5, AL ; INT1
MOV AL, 10 ; Frecuencia
OUT TIMER+1, AL ; COMP
MOV AL, 0 ; Inicio
OUT TIMER, AL ; CONT
MOV BX, OFFSET MIN
MOV AL, OFFSET FIN-OFFSET MIN
STI
LAZO: JMP LAZO
END
```

## Ejercicio 14: TIMER + F10

; Nota: PUSH y POP deben realizarse dentro del conjunto cerrado por CLI y STI

```
TIMER EQU 10H
PIC EQU 20H
EOI EQU 20H
N_F10 EQU 9
N_CLK EQU 10
```

```
ORG 36
IP_F10 DW RF10
```

```
ORG 40
IP_CLK DW RCLK
```

```
ORG 1000H
MSJ DB "Presione F10 para iniciar / pausar / reanudar"
SEG DB "30"
FIN DB ?
```

```
ORG 3000H
RF10: CLI
PUSH AX
CMP DH, 0 ; It's OFF?
JNZ OFF
MOV AL, 0FCH ; Both
INC DH ; Turn ON
MOV AL, 0
OUT TIMER, AL
JMP FF10
OFF: DEC DH ; Turn OFF
MOV AL, 0FEH ; F10 only
FF10: OUT PIC+1, AL ; Write IMR
MOV AL, EOI ; Set end
OUT PIC, AL ; Write EOI
POP AX
STI
IRET
```

```
RCLK: PUSH AX
INT 7 ; Print
DEC SEG+1 ; -1 seg
CMP SEG+1, 30H ; < X0
JNS RESET
MOV SEG+1, 39H ; X9
DEC SEG ; -10 seg
CMP SEG, 30H ; < 0X
JNS RESET
INT 0 ; Prog End
RESET: MOV AL, 0 ; Restart
OUT TIMER, AL ; CONT
MOV AL, EOI ; Set end
OUT PIC, AL ; Write EOI
```

```
POP AX
IRET

ORG 2000H
CLI
MOV AL, 0FEH ; Sólo F10
OUT PIC+1, AL ; Write IMR
MOV AL, N_F10 ; Int number
OUT PIC+4, AL ; INT0
MOV AL, N_CLK ; Int number
OUT PIC+5, AL ; INT1
MOV AL, 1 ; Frequency
OUT TIMER+1, AL ; COMP
MOV AL, 0 ; Set start
OUT TIMER, AL ; CONT
MOV DH, 0 ; Turn OFF
MOV BX, OFFSET MSJ
MOV AL, OFFSET SEG-OFFSET MSJ
INT 7
MOV BX, OFFSET SEG
MOV AL, OFFSET FIN-OFFSET SEG
STI

LAZO: JMP LAZO
END
```