

**Ejercicio 1 (18 pts.)**

a)

**Address Setup ( $T_{AS}$ ):**

Si CS\_n es la última en bajar:

$$(1) \quad T_{AS} \leq \max \{ t_{not(min)} , t_{7(min)} \} + t_{or(min)}$$

Si WE\_n es la última en bajar:

$$(2) \quad T_{AS} \leq T_1 + T_{2H} + t_{30(min)} - t_{6(max)}$$

**Data Setup ( $T_{DS}$ ):**

$$(3) \quad T_{DS} \leq T_{1L} + T_2 + nT_W + T_{3H} + \min \{ ( t_{12(min)} + t_{or(min)} ) , t_{32(min)} \} - t_{53(max)}$$

**b) Para el sistema especificado:**

$$N = 0$$

$$T = 1/F_{clk} = 167 \text{ ns}$$

$$T_L = T_H = T/2 = 83 \text{ ns}$$

$$t_{53(max)} = 130 \text{ ns}$$

$$t_{32(min)} = 0$$

$$t_{30(min)} = 0$$

$$t_{12(min)} = 0$$

$$t_{6(max)} = 90 \text{ ns}$$

$$t_{6(min)} = 0 \text{ ns}$$

$$t_{7(min)} = 35 \text{ ns}$$

Para este sistema lo más \*temprano\* que puede llegar a bajar WE\_n respecto a la subida de T1 es:

$$t_{we\_fall} = T_1 + T_{2H} + t_{30min} = 250 \text{ ns}$$

Lo más \*tarde\* que puede llegar a bajar CS es:

$$t_{cs\_fall} = \max \{ t_{6max} + t_{notmax}, T_{1H} + t_{8max} \} + t_{ormax} = 165 \text{ ns}$$

Por lo tanto WE\_n siempre es la última en bajar así que nos interesa solo (2):

$$T_{AS} \leq T_1 + T_{2H} + t_{30(min)} - t_{6(max)} = 250 \text{ ns} + 0 \text{ ns} - 90 \text{ ns}$$

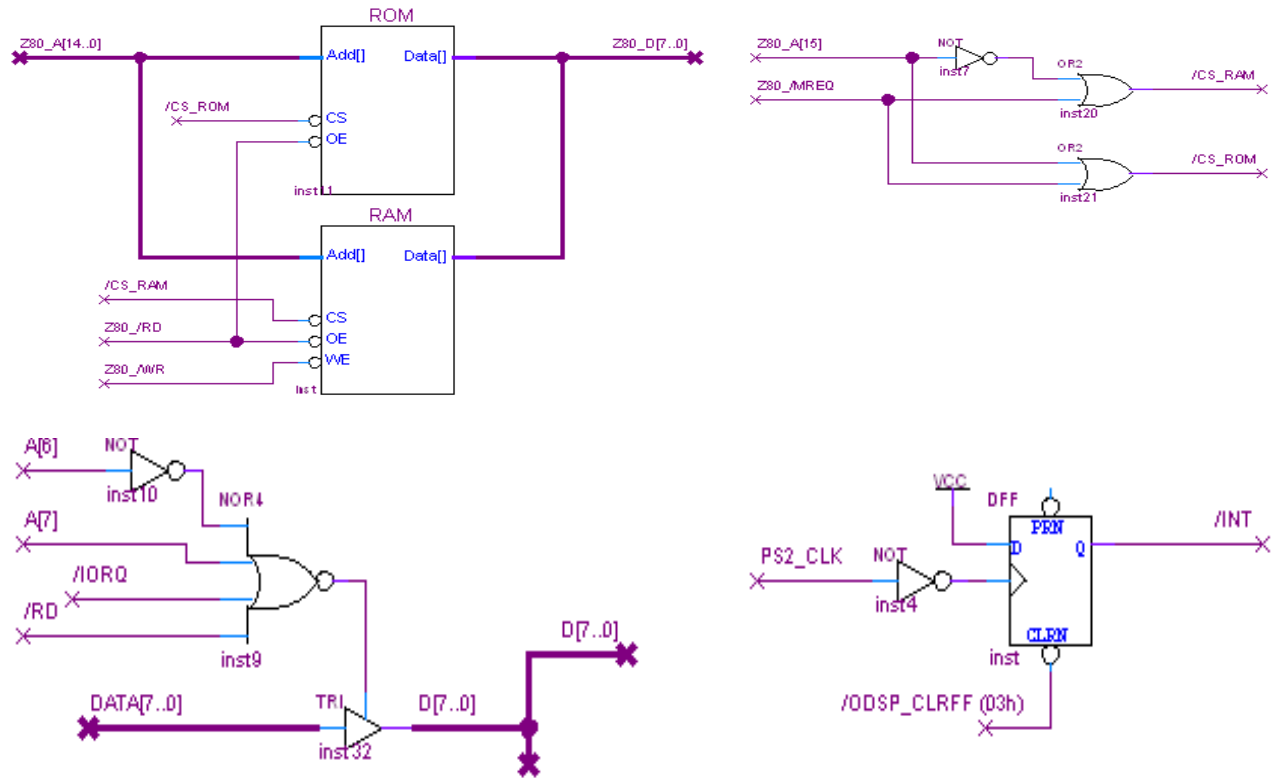
$$\mathbf{T_{AS} \leq 160 \text{ ns}}$$

y (3):

$$T_{DS} \leq T_{1L} + T_2 + T_{3H} - \min \{ ( t_{12(min)} + t_{or(min)} ) , t_{32(min)} \} - t_{53(max)}$$

$$\mathbf{T_{DS} \leq 203 \text{ ns}}$$

**Ejercicio 2 (26 pts.)**  
**a) Hardware**



**b) Software**

```

PSDATA          equ    0x40

org 0x0000
init:
    LD SP, 0      ;; inicialización
    IM 1
    LD A, 0
    LD (BITPOS), A
    CALL init_otros
    EI
    JP LOOP      ;; loop principal

;; variables
org 0x8000
BITPOS: DB      ;; {0,10} 0=start 10=stop
RXDATA: DB      ;; dato parcialmente recibido

;; rutina interrupción
org 0x0038
ISR:
    PUSH AF
    PUSH BC
    IN A, (PSDATA)
    AND 0x01
    LD C, A      ; bit recibido en LSB de C

;; BITPOS dice cual de los 11 bits serie
;; acabo de recibir
    LD A, (BITPOS)
    CP 0        ; BITPOS==0 : start
    JP Z, START_BIT
    CP 8        ; 0<BITPOS<=8 : datos
    JP M, RX
    CP 9        ; BITPOS==9 : parity
    JP Z, RX_PARITY
    JP STOP_BIT ; else: stop

START_BIT:

```

```

LD A, 0
LD (RXDATA), A ; init rx parcial
JP NEXT      ; nada mas que hacer

RX:
LD A, (RXDATA)
OR C
RR A
LD (RXDATA), A
JP NEXT

RXPARITY:
;; copio dato recibido
LD A, (RXDATA)
LD (DATO), A
;; copio paridad recibida
LD A, C
LD (PARIDAD),A

;; aviso que hay un nuevo dato
LD A, 0xFF
LD (HAY_DATO), A
JP NEXT

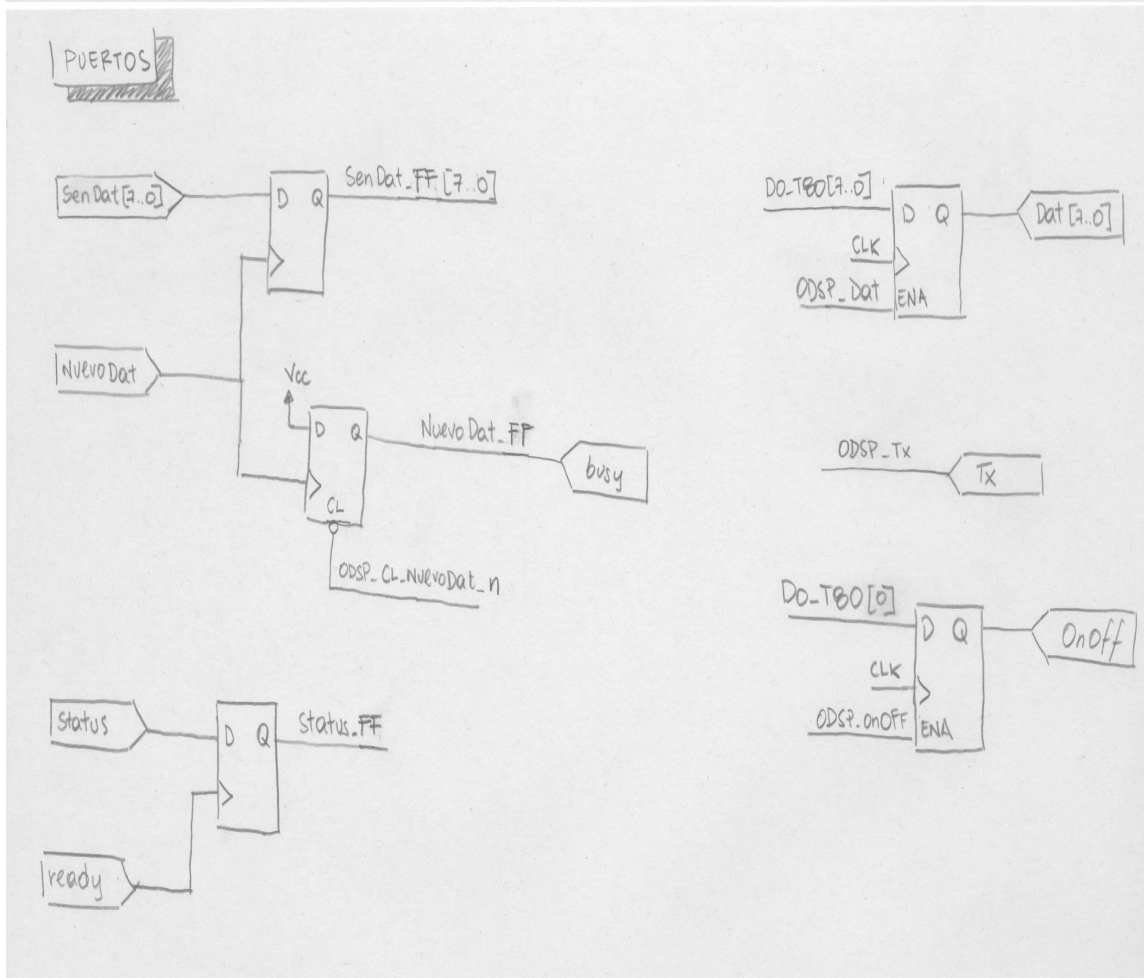
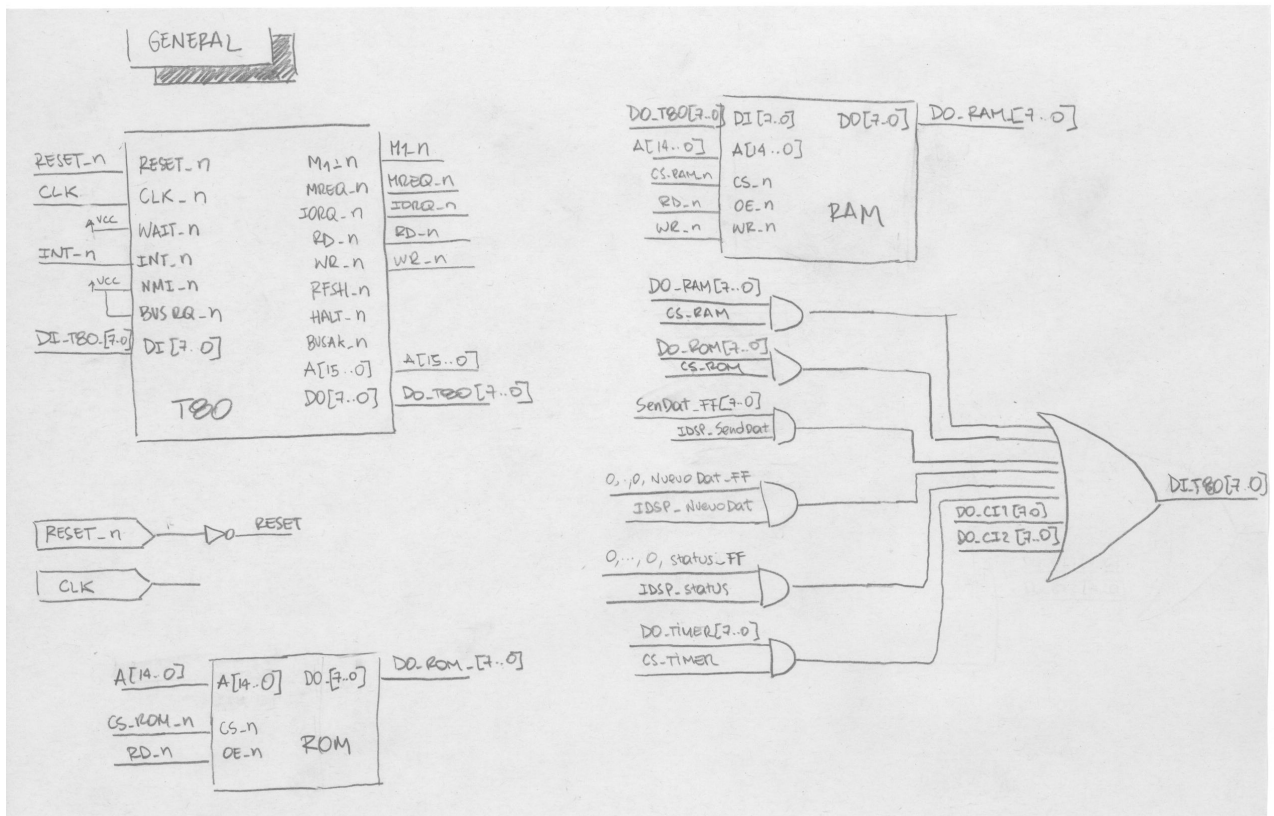
STOP_BIT:
;; reseteo BITPOS para la próxima
LD A, 0
LD (BITPOS), A
JP DONE ; fin de la recepción

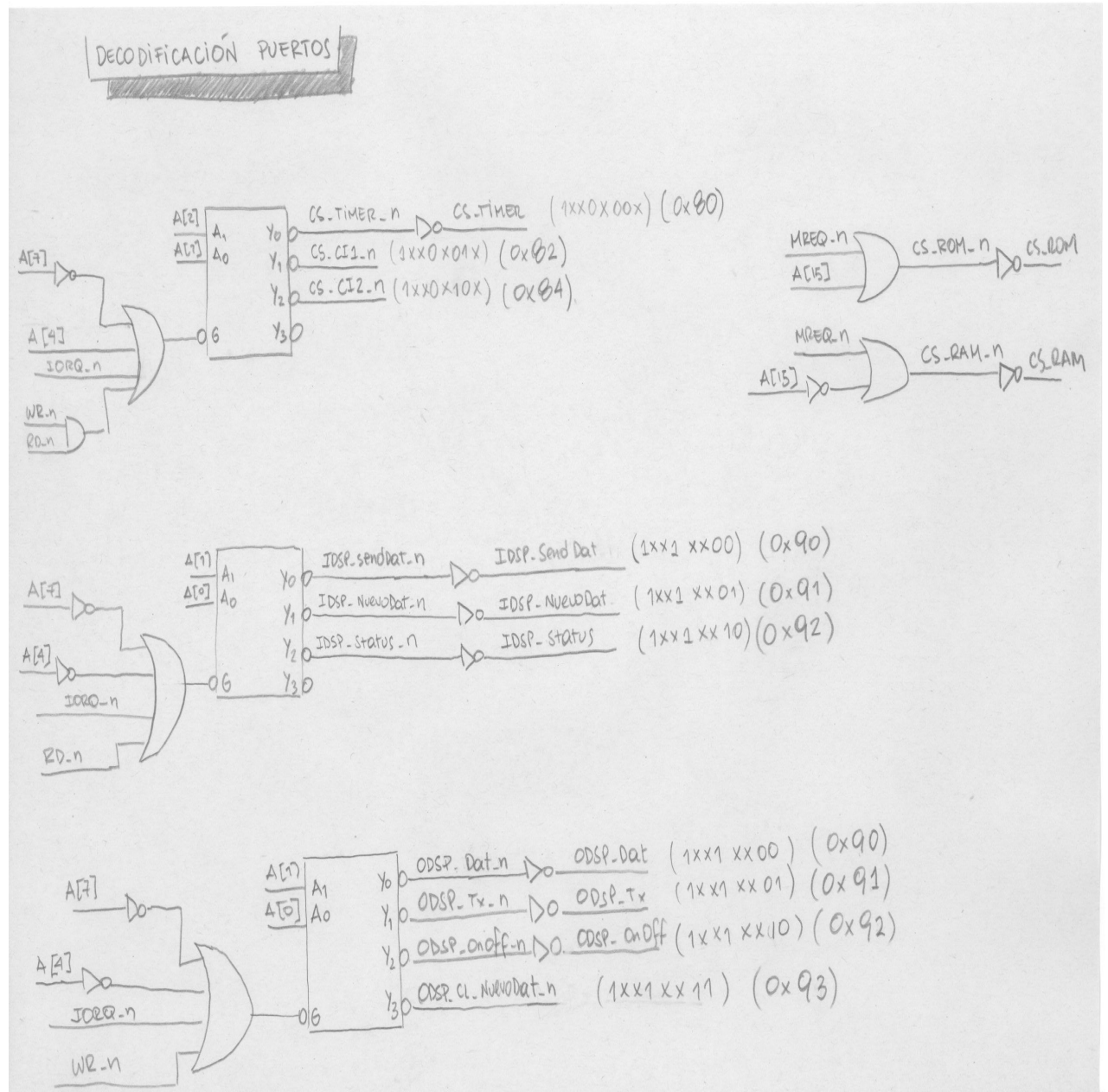
NEXT:
;; listo para recibir siguiente bit
LD A, (BITPOS)
INC A
LD (BITPOS), A

DONE:
POP BC
POP AF
EI
RET

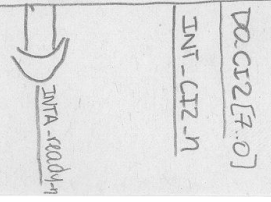
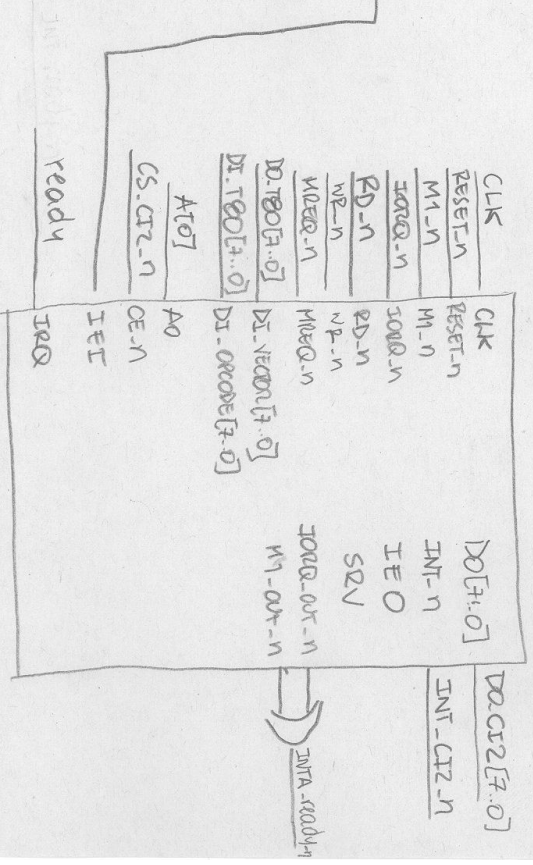
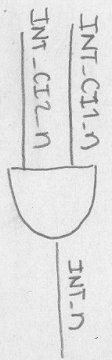
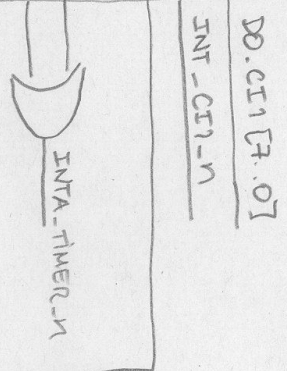
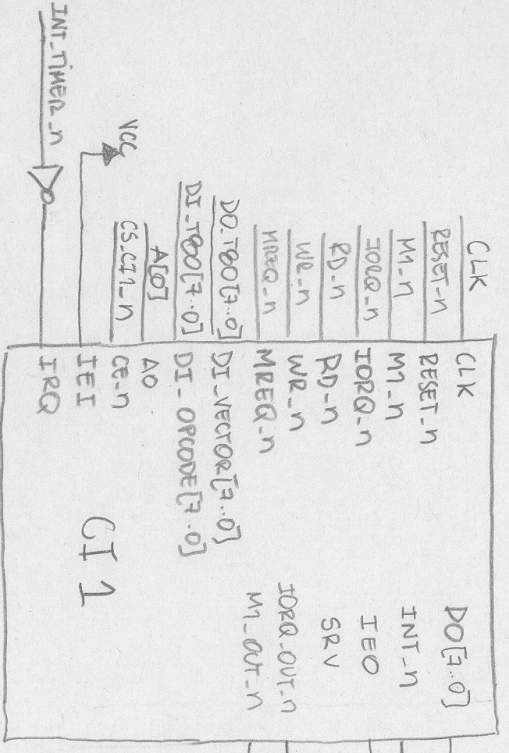
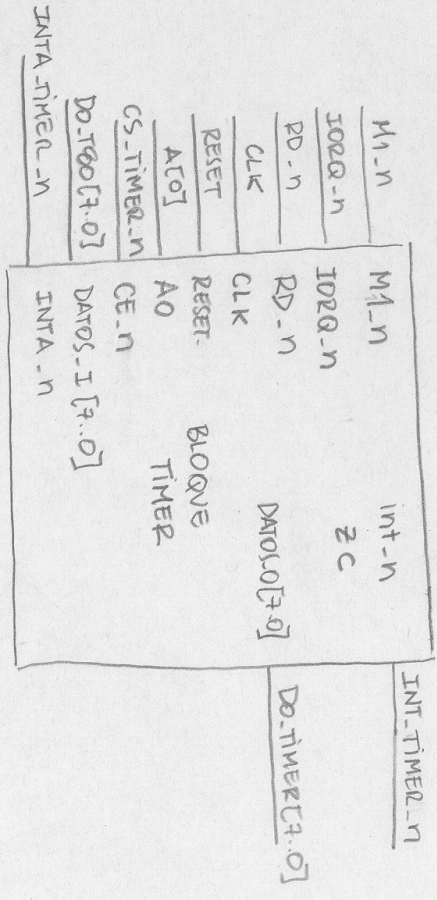
```

Ejercicio 3 (56 pts.)





INTERRUPCIONES





```

CALL getCola                ;Saco un dato de la cola, lo devuelve en A
JP NZ, haydato              ;Si hay dato voy a transmitir
JP fin_rutint_timer         ;Si no hay dato terminé, como el timer esta
                             ;periódico no precisa reconfiguración

haydato:
  OUT(Dat), A                ;Escribo el dato en puerto de salida
  LD A, RADIO_ON             ;Prendo la radio
  OUT (OnOff), A
  OUT (TX), A                ;Genero flanco en TX
fin_rutint_timer:
  POP AF                     ;Restauro registros
  RETI                       ;Retorno
;; Rutina de la radio
; Pseudocódigo
; Leo estado
; if estado=exito
;   Leo cola
;   if cola= vacia
;     apago
;   else
;     coloco dato en puerto de salida
;     genero pulso en tx
;     fin
;   end
; else
;   genero pulso en tx (como es reenvío el dato ya esta en el puerto de salida)
;   fin
; end
;
rutint_radio:
  EI                         ;Habilito interrupciones
  PUSH AF                    ;Preservo registros
  IN A, (status)             ;Me fijo el estado de la ultima transmisión
  JP Z, reenvio              ;Si no fue exitosa voy a reenvío
                             ;Si fue exitosa
envio:
  CALL getCola               ;Traigo un nuevo dato de la cola
  JP Z, apago                ;Si la cola esta vaciá apago y salgo, sino
                             ;envío.
  OUT(Dat), A                ;Escribo el dato en el puerto
  OUT(Tx), A                 ;Genero flanco en Tx
  JP fin_rutint_radio        ;Me voy
reenvio:
                             ;Si hay que reenviar, el dato ya esta en el
                             ;puerto y alcanza con solo dar el flanco en TX
  OUT (Tx), A                ;Genero flanco en Tx
  JP fin_rutint_radio        ; Me voy
apago:
  LD A, RADIO_OFF            ;Si esta vaciá apago la radio y termine
  OUT (OnOff), A
fin_rutint_radio:
  POP AF                     ;Preservo Registros
  RETI                       ;Retorno

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  ORG 0x0600                  ;Tabla de interrupciones alineada en ROM
tabla_int:
  DW rutint_timer
  DW rutint_radio
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Reserva memoria
ORG 0x8000                    ;Variables en RAM
Cola: DS 256                  ;La cola de 256 bytes

```