

# Tarea 5 M: Cro procesadores

Bryan Cortés Espinola C22422

## Memoria de Calculo

Para la integración output compare en el canal 4

$$T_{OC} = \frac{PRS \times T_{C4}}{BUS\_CLK}, \text{ se desea una frecuencia } f_{OC} = 50 \text{ KHz}$$
$$\Rightarrow T_{OC} = \frac{1}{f_{OC}} = 20 \text{ ns}$$

Al despejar  $T_{C4}$

$$T_{C4} = \frac{T_{OC} \times BUS\_CLK}{PRS}$$

El valor de  $BUS\_CLK$  en la dragon 12 es de 24 MHz, y si se toma  $PRS = 16$

$$T_{C4} = \frac{20 \text{ ns} \times 24 \text{ MHz}}{16} = 30$$

## Tareas y sus rutinas de Software:

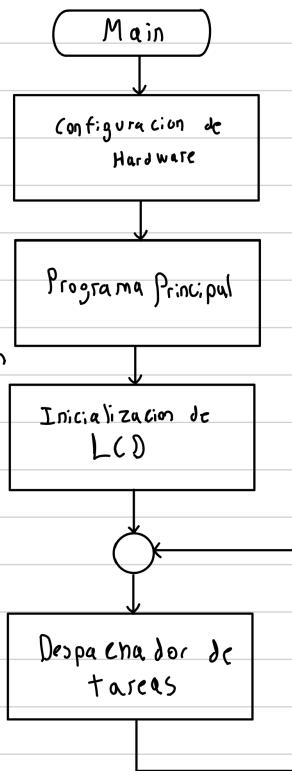
Main: El programa Main lo Componen las siguientes partes

Configuración de Hardware: Configura y habilita el hardware a utilizar, como los LEDs, Display de 7 segmentos, etc configura el output compare y habilita la interrupción y el puerto A para el teclado matricial

Programa Principal: se inicializa las variables y timers a utilizar en el programa, así como el puntero de pila, habilita interrupciones y los estados presentes de las tareas.

Inicialización del LCD: Configura la pantalla LCD, la inicializa y borra su contenido, dejando la pantalla lista para su uso en el programa.

Despachador de tareas: se ejecuta ciclicamente los tareas del programa



# Configuración de HardWare

Estructuras de datos

Carga\_TC4: Valor de la cuenta a  
llegar del OC para una frecuencia  
de 50KHz

Configuración del  
Hardware

DDR\_B ← \$FF  
DDRJ\_1 ← 1  
PTJ\_1 ← 1

Leds

DDR\_P ← \$7F  
PTP ← \$07

Displays

TC4 ← Carga\_TC4  
TSCR1 ← \$90  
TSCR2 ← \$04  
Tios ← \$10  
Tie ← \$10

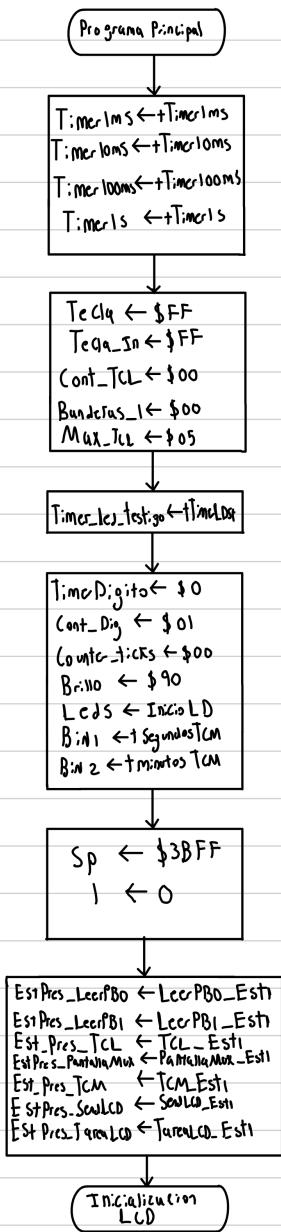
Output Compare

RR1 ← (TCNT)  
RR1 ← RR1 + Carga\_TC4  
TC4 ← RR1

Iniciar cuenta  
de OC

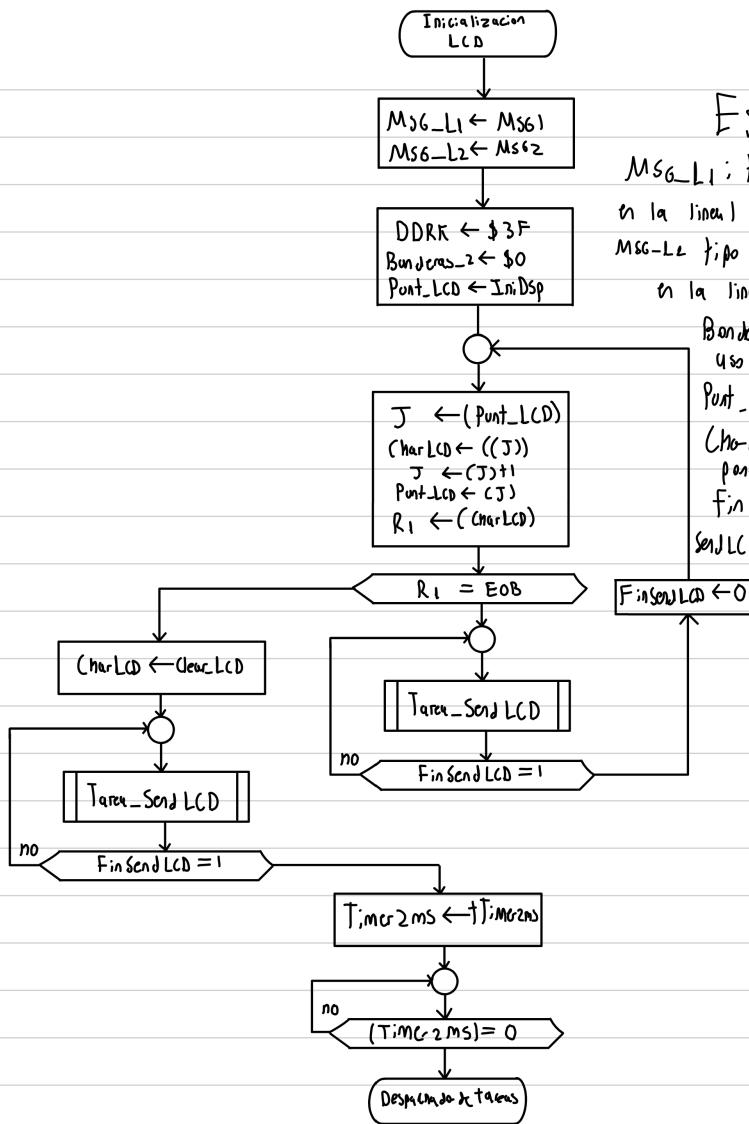
DDRA ← \$F0  
PUCR\_1 ← 1

Programa Principal



### Estructura de datos

**Timer1ms:** Variable tipo Word, timer para la buse 1ms  
**Timer10ms:** Variable tipo Word, timer para la buse 10ms  
**Timer100ms:** Variable tipo Word, timer para la buse 100ms  
**Timer1s :** Variable tipo Word, timer para la buse 1s  
**Timer\_led\_testigo :** Variable tipo byte, Para el tiempo del led testigo  
**Time-Digito :** tipo byte Variable para el periodo de cada digito mostrado  
**TeClq :** variable tipo byte ; guarda una tecla ingresada  
**TeClq\_in:** variable tipo byte ; guarda una tecla ingresada  
**Cont\_TCL :** variable tipo byte , Contiene la cantidad de teclas ingresadas  
**Banderas\_1 :** variable tipo byte , Contiene los banderas asociadas al boton y teclado  
**MAX\_TCL :** variable tipo byte, maximo numero de teclas ingresadas  
**Cont\_dig :** tipo byte , indica cual display poner el digito  
**Counter\_ticks:** tipo word, Timer para controlar el brillo de los led y displays  
**Brillo :** tipo byte , Controla el brillo a llegar  
**Leds :** tipo byte , contenido a escribir en los led's  
**Bin1 :** tipo byte , contenido a escribir en los displays 3 y 4  
**Bin2 :** tipo byte , contenido a escribir en los displays 1 y 2  
**Est\_Pres\_X:** tipo word, guarda la dirección del estado presente



Estructuras de datos

MSG\_L1 : tipo Word, puntero del mensaje a poner en la linea

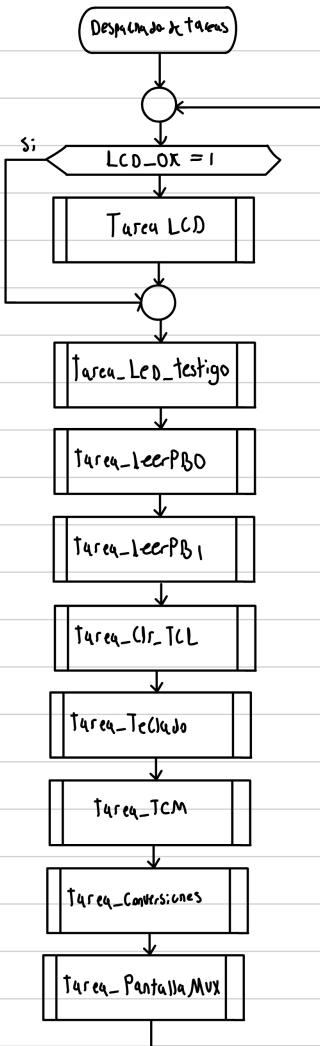
MSG\_L2 : tipo Word, puntero del mensaje a poner en la linea 2

Bandera\_2 : tipo byte, contiene los banderas usos Unidos al LCD

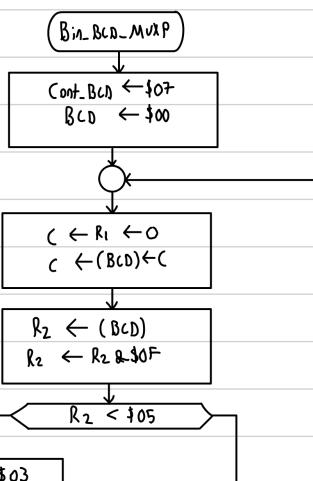
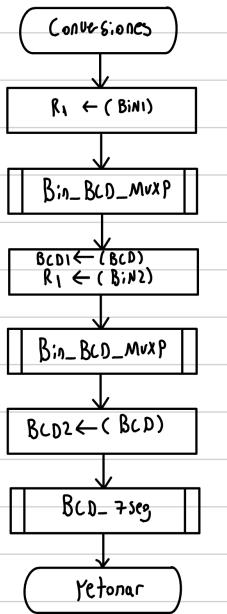
Punt\_LCD : tipo word, puntero para el LCD

CharLCD : tipo word, puntero del carácter a poner en pantalla

FinSendLCD : Bandera, indica si la tarea SendLCD terminó de mandar el carácter.

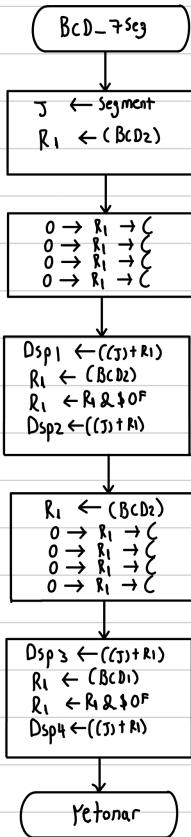


# Tarea Conversiones: Convertir un valor Binario a un valor para mostrar en el display de siete segmentos



Estructuras de Datos

- Bin1 : Variable tipo byte
- Bin2 : Variable tipo byte
- BCD : variable tipo byte
- BCD1 : Variable tipo byte
- BCD2 : Variable tipo byte



## Estructuras de Datos

Bin1 : Variable tipo byte

Bin2 : Variable tipo byte

BCD : Variable tipo byte

BCD1 : Variable tipo byte

BCD2 : Variable tipo byte

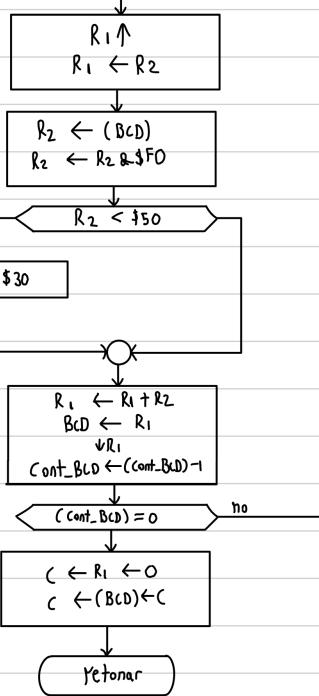
(Cont\_BCD) : Variable tipo byte

Segmento: dirección de la tabla de los

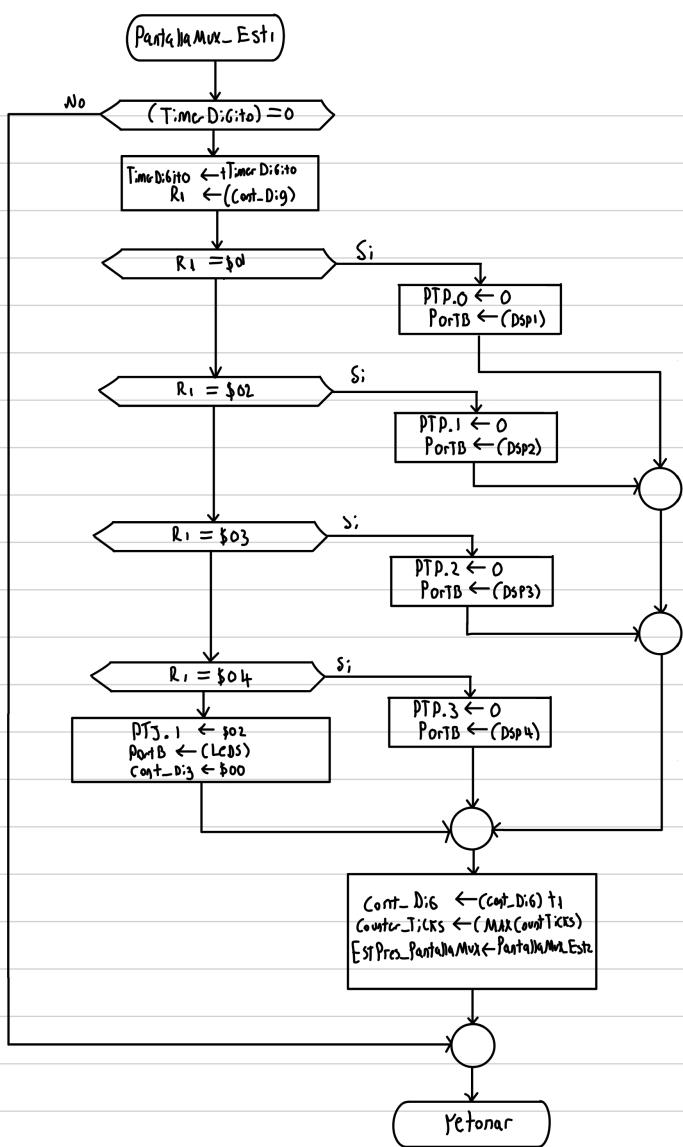
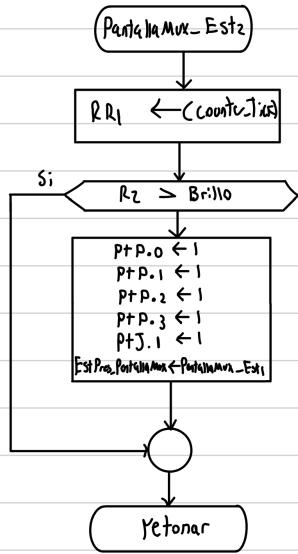
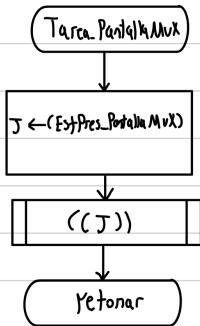
valores de 7 segmentos

Dspx : Variables tipo byte guarda lo que se

desea mostrar en el display



Tarea Pantalla Mux : Muestra valores en los displays



Estructuras de datos

EstPres\_PantallaMux : tipo word

DSPx : variables byte

Counter\_ticks : variable word

TimeDigito : tipo byte

Cont\_Dig : tipo byte

Tarea Send LCD; Manda datos a la Pantalla LCD por medio de protocolo paralelo strobe

SendLCD\_Esti

```

 $R_1 \leftarrow (charLCD)$ 
 $R_1 \leftarrow R_1 \& \$FO$ 
 $O \rightarrow R_1 \rightarrow c$ 
 $O \rightarrow R_1 \rightarrow c$ 
PortKC  $\leftarrow R_1$ 

```

Estructura de datos

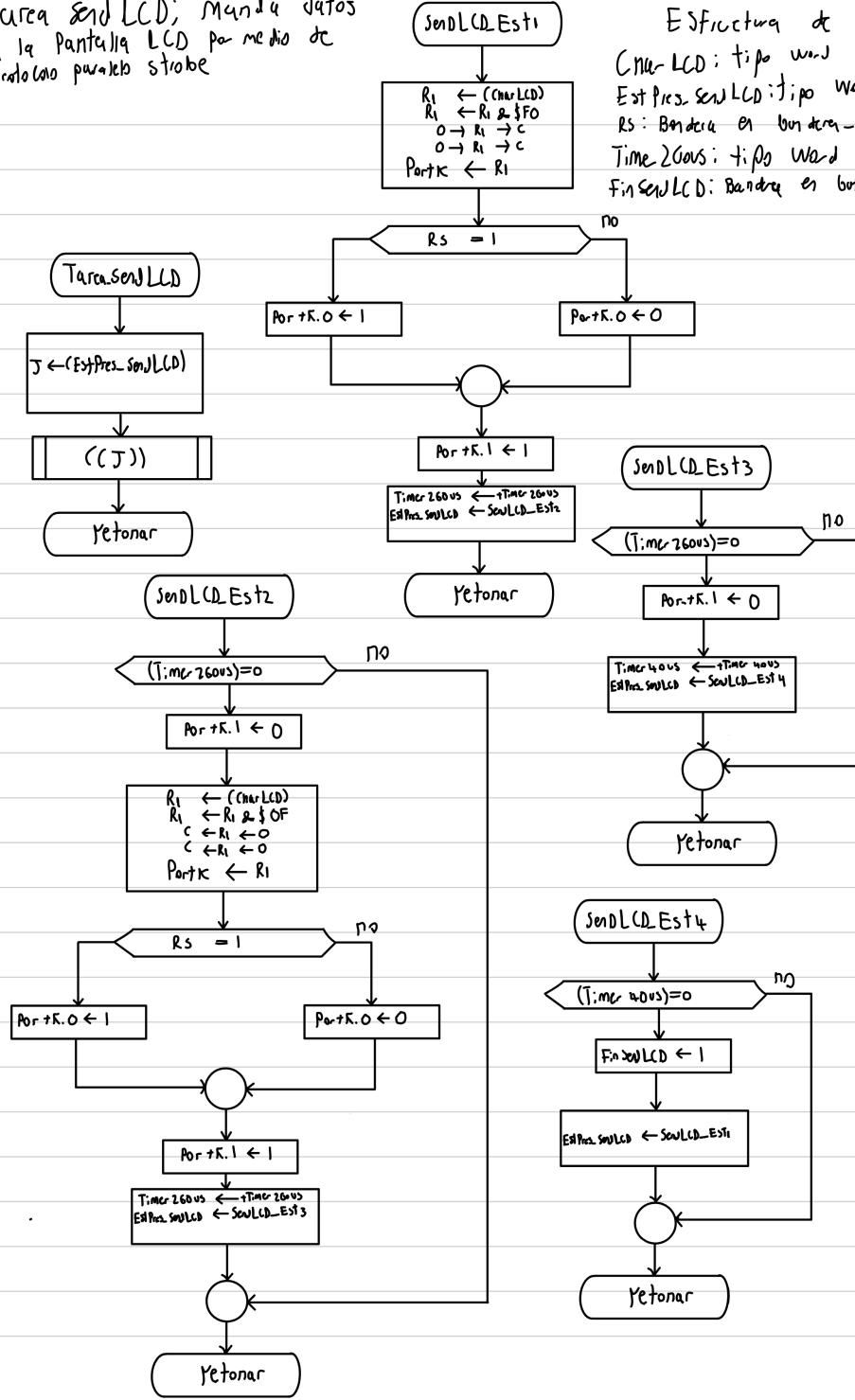
CharLCD : tipo word

EstPres\_SentLCD : tipo word

RS : Bandera en bivalente-2

Time260us : tipo word

FinSentLCD : Bandera en bivalente-2



Tarea LCD, Controla la pantalla LCD para desplegar los mensajes

### Estructura de datos

EstPres\_TareaLCD: Variable tipo Word

FisegnLCD: Bandera

Second\_line: Bandera

LCD\_OR: Bandera

(char\_C): tipo byte

Punt\_LCD: tipo Word

Tarea LCD

J ← EstPres\_TareaLCD  
.

((J))

Retornar

TareaLCD\_Est2

FinSendLCD = 0

FinSendLCD ← 0  
RS ← 1  
J ← (Punt\_LCD)  
CharLCD ← ((J))  
J ← (J) + 1  
Punt\_LCD ← J  
R1 ← (CharLCD)

no

R1 = EOB

s:

Second\_Line = 1

Second-Line ← 0  
LCD\_OR ← 1

EstPres\_TareaLCD ← TareaLCD\_Est

Retornar

TareaLCD\_Est1

FinSendLCD ← 1  
RS ← 1

Second\_line = 1

s:

CharLCD ← ADD\_L1  
Punt\_LCD ← (Msg\_L1)

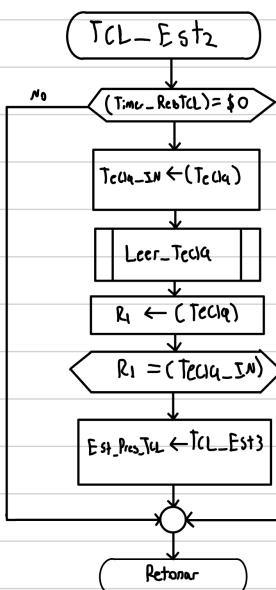
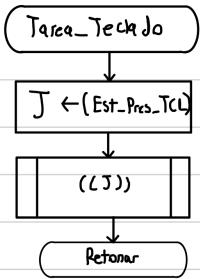
CharLCD ← ADD\_L2  
Punt\_LCD ← (Msg\_L2)

Tarea\_SendLCD

EstPres\_TareaLCD ← TareaLCD\_Est

Retornar

Tarea\_Teclado : Gestiona el ingreso de teclas  
del teclado matricial



Estructura de Datos

Est\_Pres\_TCL : tipo word

Timer\_ResTCL : tipo byte

Tecla : tipo byte

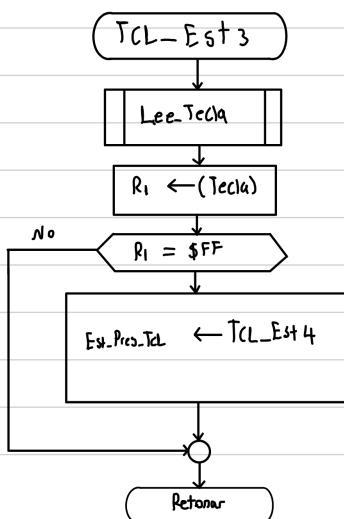
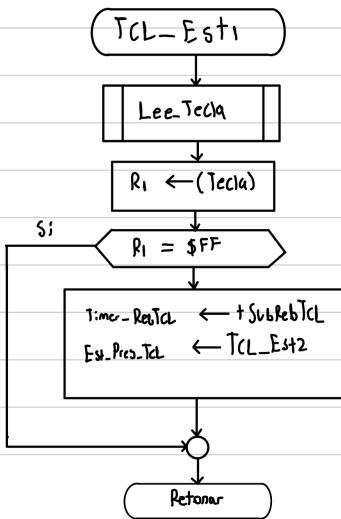
Tecla\_In : tipo byte

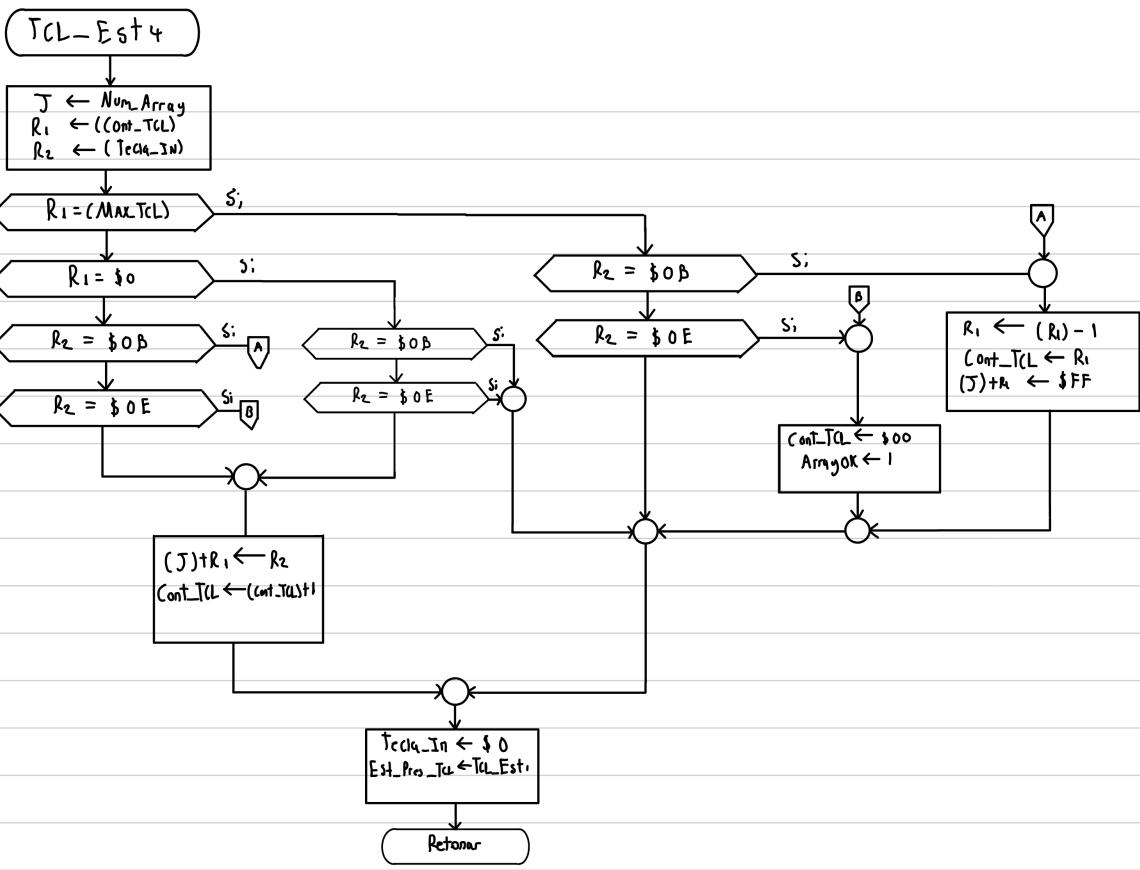
Cont\_TCL : tipo byte

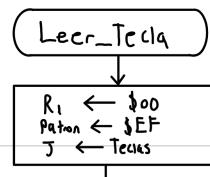
Max\_TCL : tipo byte

num\_array ; dirección donde se guardan las teclas validas

array\_OK : Bandera







Sus rutinas y teciles asociadas al teclado

Sus rutina Leer\_Tecila: Lee si alguna tecla fue presionada y se guarda en la variable tecila

Borrar\_Num\_Array: Subrutina el cual borra la memoria de tecclas ingresados

Tarea\_CIR\_TCL: Verifica si se presionó el botón de clear para borrar num\_array

Estructuras de datos

Patron; tipo byte

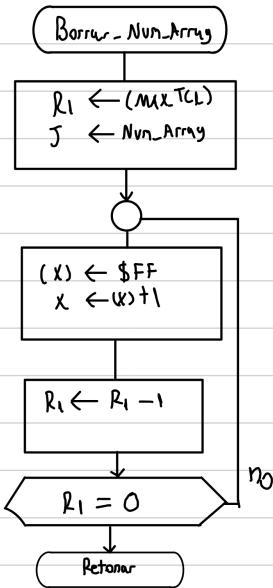
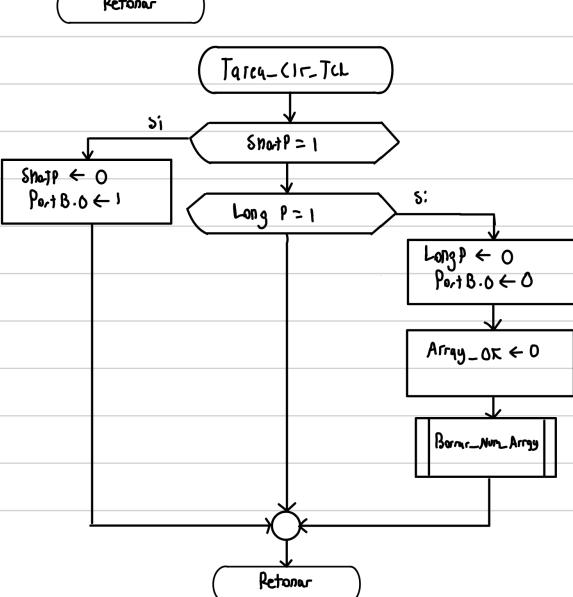
tecla; tipo byte

Max\_TCL; tipo byte

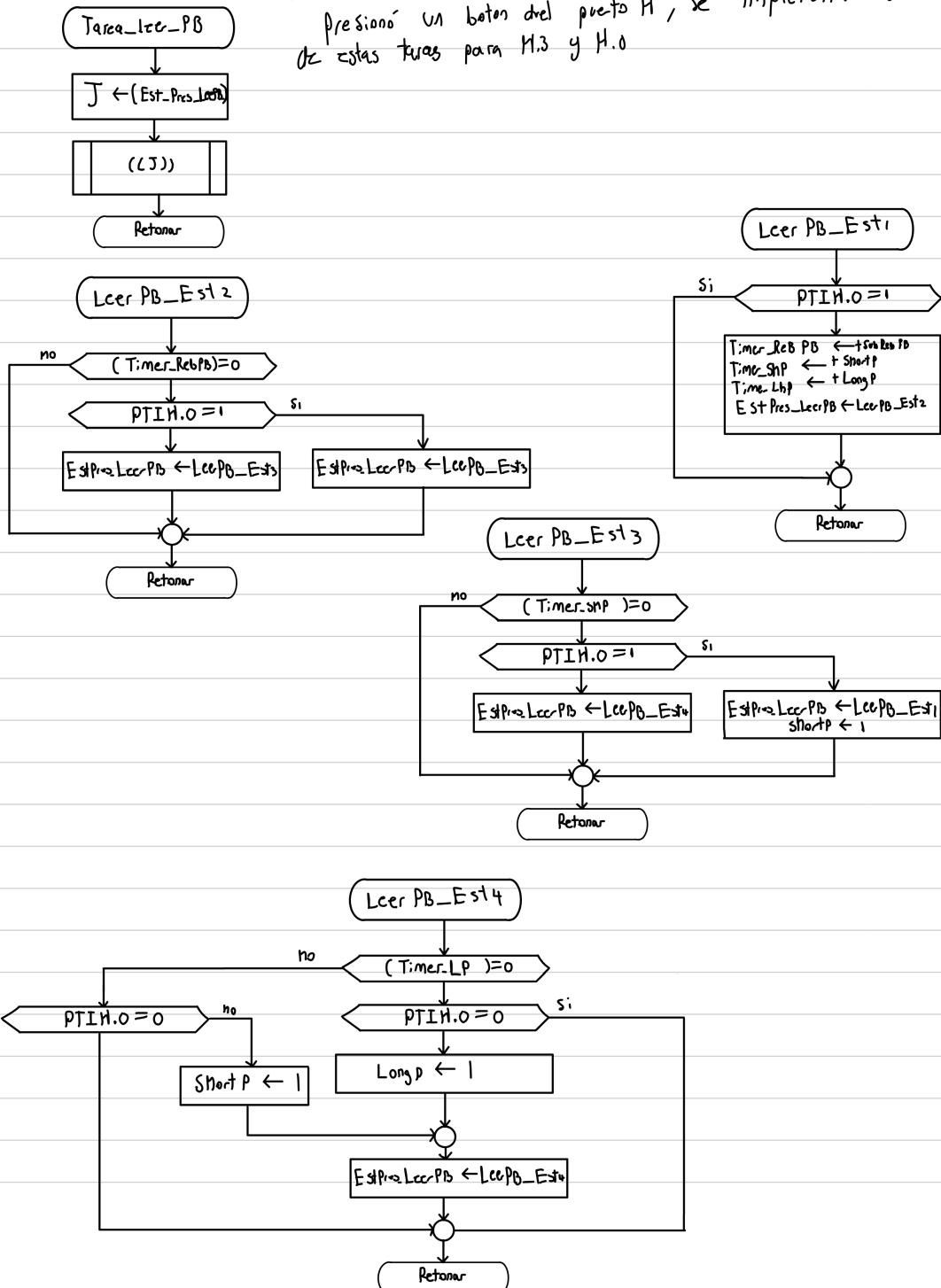
ShortP; Bandera

LongP; Bandera

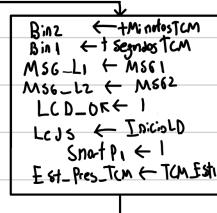
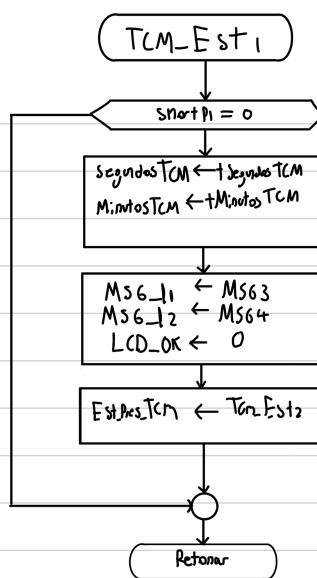
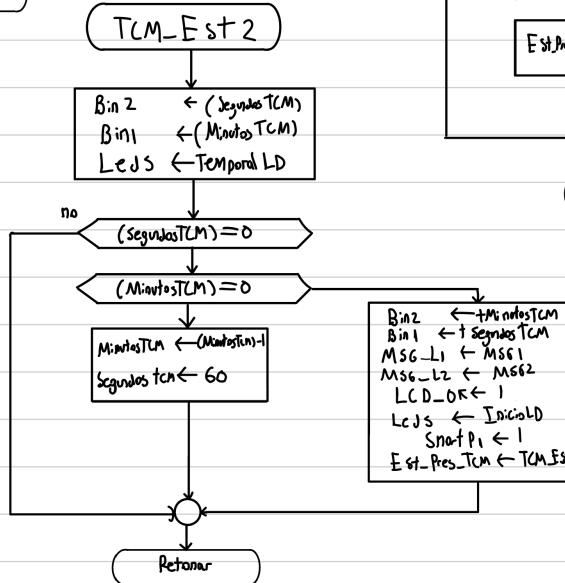
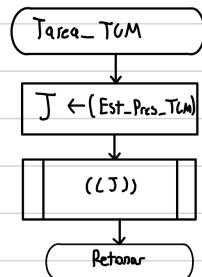
Array\_OF; Bandera



Tarea Lee\_PB : se encarga de ver si se presionó un botón del preto H, se implementan dos de estas tareas para H.3 y H.0



Tarea\_TCM: Implementa El reloj mostrado en los displays de 7 segmentos



Tarea led testigo: se modificó la tarea y se colocó en el led tricolor implementado un estado para cada color de led

Tarea\_LedTestigo

$J \leftarrow (\text{Est\_Pres\_Dato})$

( $J$ )

Retornar

LDTst\_Est1

Timer\_Led\_Testigo  $\geq 0$   
PTP.4  $\leftarrow 1$   
PTP.6  $\leftarrow 0$   
Est\_Pres\_Dato  $\leftarrow \text{Ldtst\_Est1}$   
Time\_LedTestigo  $\leftarrow$  Timer\_LDtst

Retornar

LDTst\_Est2

Timer\_Led\_Testigo  $\geq 0$

PTP.4  $\leftarrow 0$   
PTP.5  $\leftarrow 1$   
Est\_Pres\_Dato  $\leftarrow \text{Ldtst\_Est2}$   
Time\_LedTestigo  $\leftarrow$  Timer\_LDtst

Retornar

LDTst\_Est3

Timer\_Led\_Testigo  $\geq 0$

PTP.5  $\leftarrow 0$   
PTP.6  $\leftarrow 1$   
Est\_Pres\_Dato  $\leftarrow \text{Ldtst\_Est3}$   
Time\_LedTestigo  $\leftarrow$  Timer\_LDtst

Retornar

Subrutina de interrupcion OC  
Atiende la subrutina y decrementa  
los timer correspondientes

