

# Sistemas Digitales: Introducción a los microcontroladores Parte 3

Ingeniería Electrónica  
UPC 2018

Por Kalun Lau

# Ejercicio: Auto Fantástico KITT

Ejercicio:

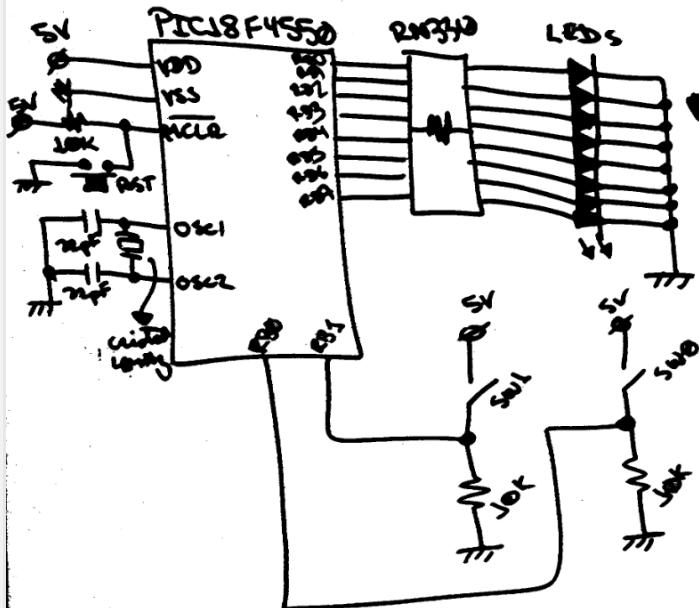
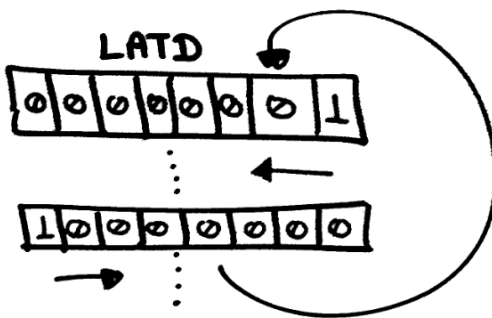
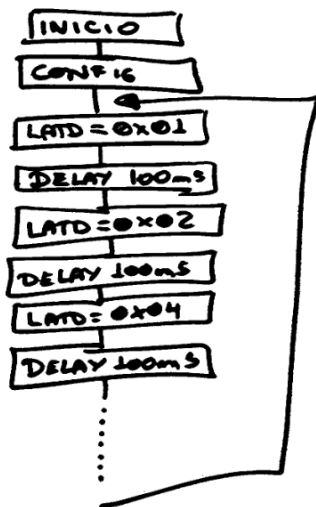


Diagrama de Flujo:



Programa "Luz":



!! Buenas tardes  
Profesor Kalun !!

```
cblock 0x0020
    cta_a
    cta_b
    cta_c
endc

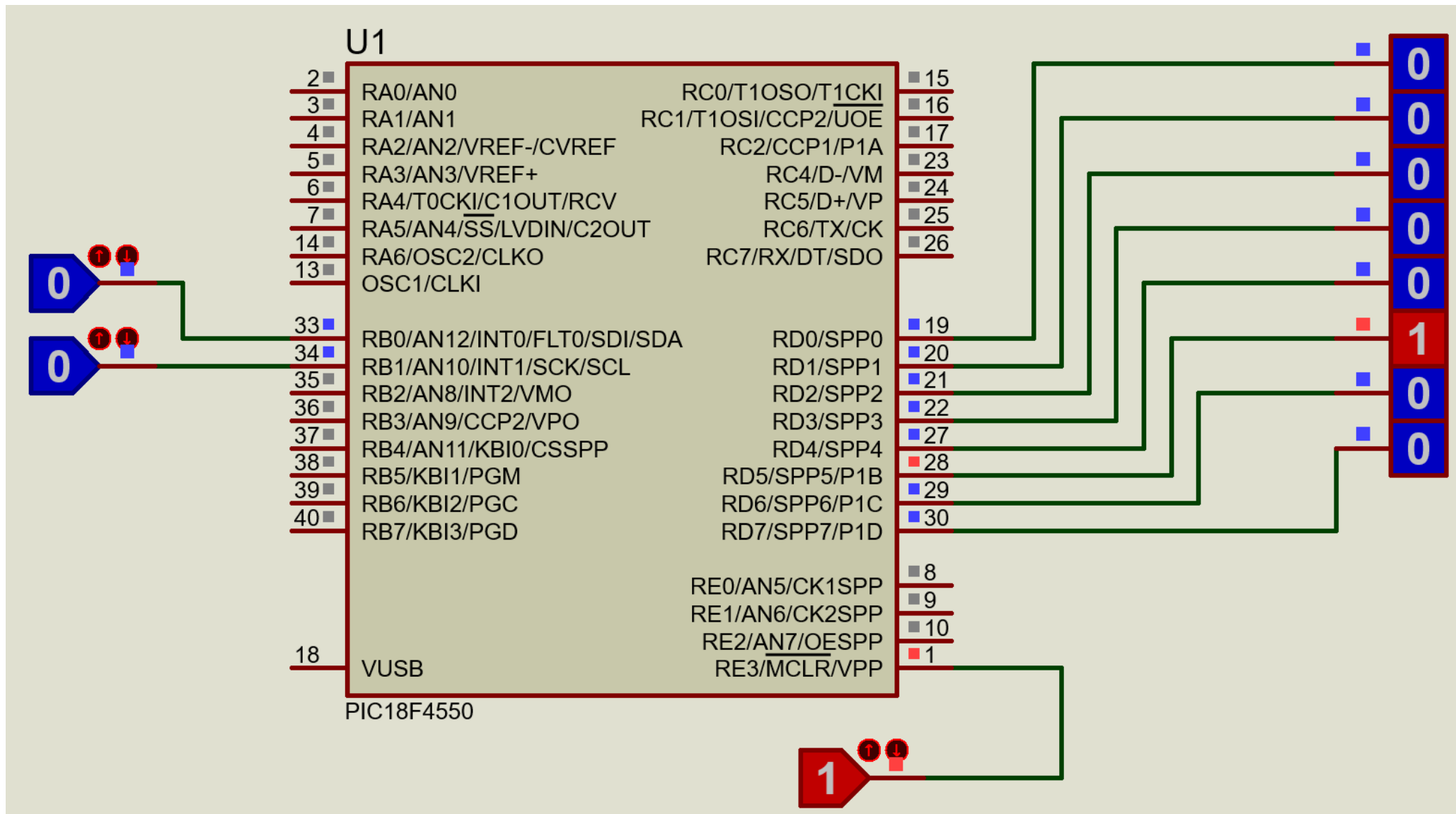
org 0x0000
goto configura

org 0x0020
configura:
    clrf TRISD

inicio:
    movlw 0x01
    movwf LATD
    call delaymon
    movlw 0x02
    movwf LATD
    call delaymon
    movlw 0x04
    movwf LATD
    call delaymon
    movlw 0x08
    movwf LATD
    call delaymon
    movlw 0x10
    movwf LATD
    call delaymon
    movlw 0x20
    movwf LATD
    call delaymon
    movlw 0x40
    movwf LATD
    call delaymon
    goto inicio
```

Nota: Se han recortado  
la zona inicial de librerías  
y bits de configuración  
- La rutina de retardo es  
la misma del ejemplo anterior

# Ejercicio: Auto Fantástico KITT



# Ejercicio: Auto Fantástico KITT

**clr f [reg]**

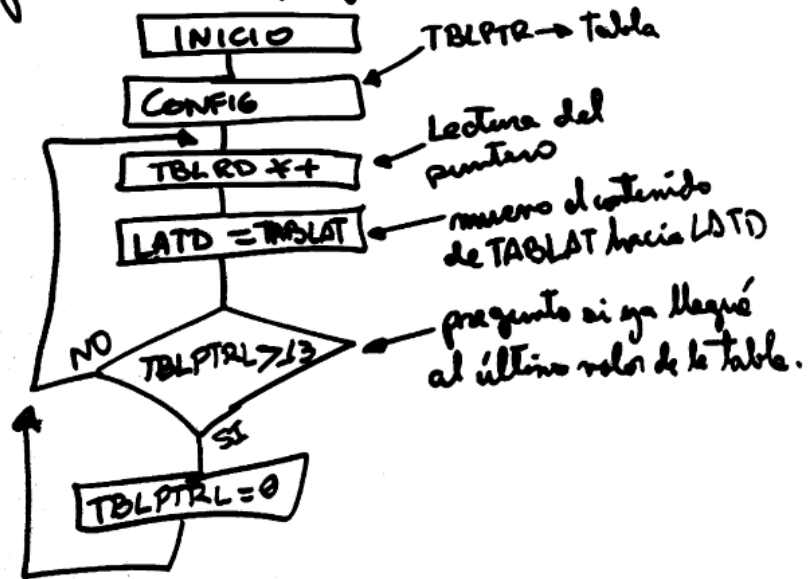
coloca todos los bits de [reg] en cero.

**setf [reg]**

complemento de "clr f"

¡El programa "lazy" ocupa mucho espacio!

Aplicando el "pagüer":



¿Cómo se escriben los números?

Decimal: 100

d'100'

Hexadecimal: 0x55

Binario b'1010100'

TBLPTR

TBLPTRU	TBLPTRH	TBLPTRL
---------	---------	---------

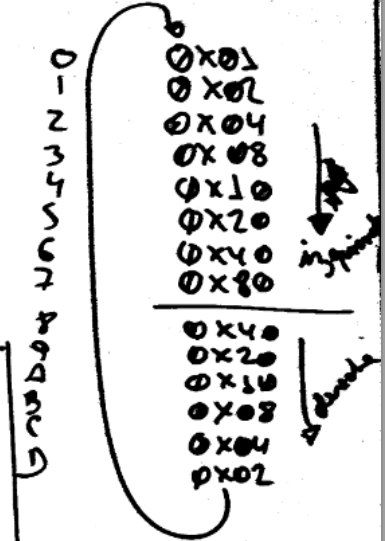
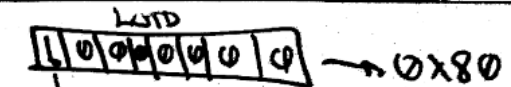
TBLRD \* +

Read  
Resultado  
TABLAT

**cpfsgt [reg]**  
F — greater than  
Pregunta [reg] > W

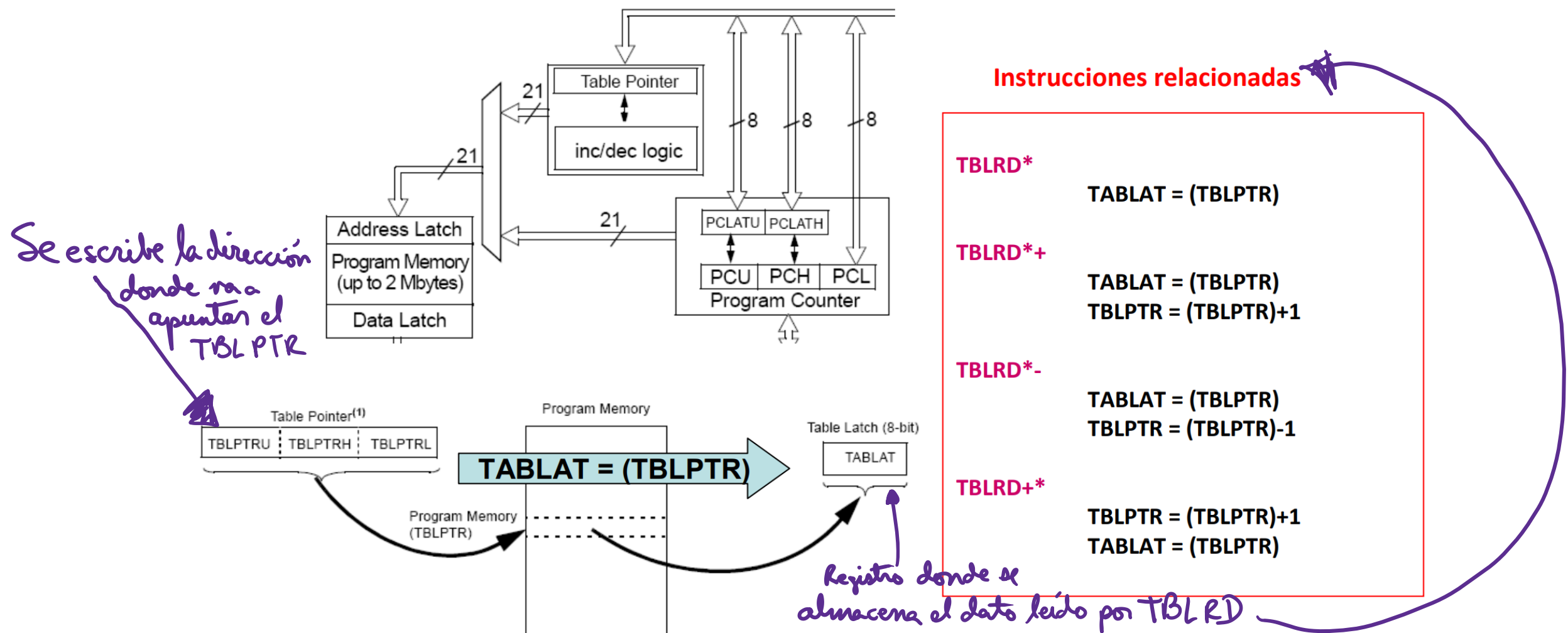
**cpfseg [reg]**  
F — equal  
Pregunta [reg] = W

**cpfslt [reg]**  
F — less than  
Pregunta [reg] < W



# Acceso a los datos almacenados en la memoria de programa

**Direccionamiento de datos (memoria programa):** Tablas datos en memoria de programa (PUNTEROS)



# Ejercicio: Auto Fantástico KITT

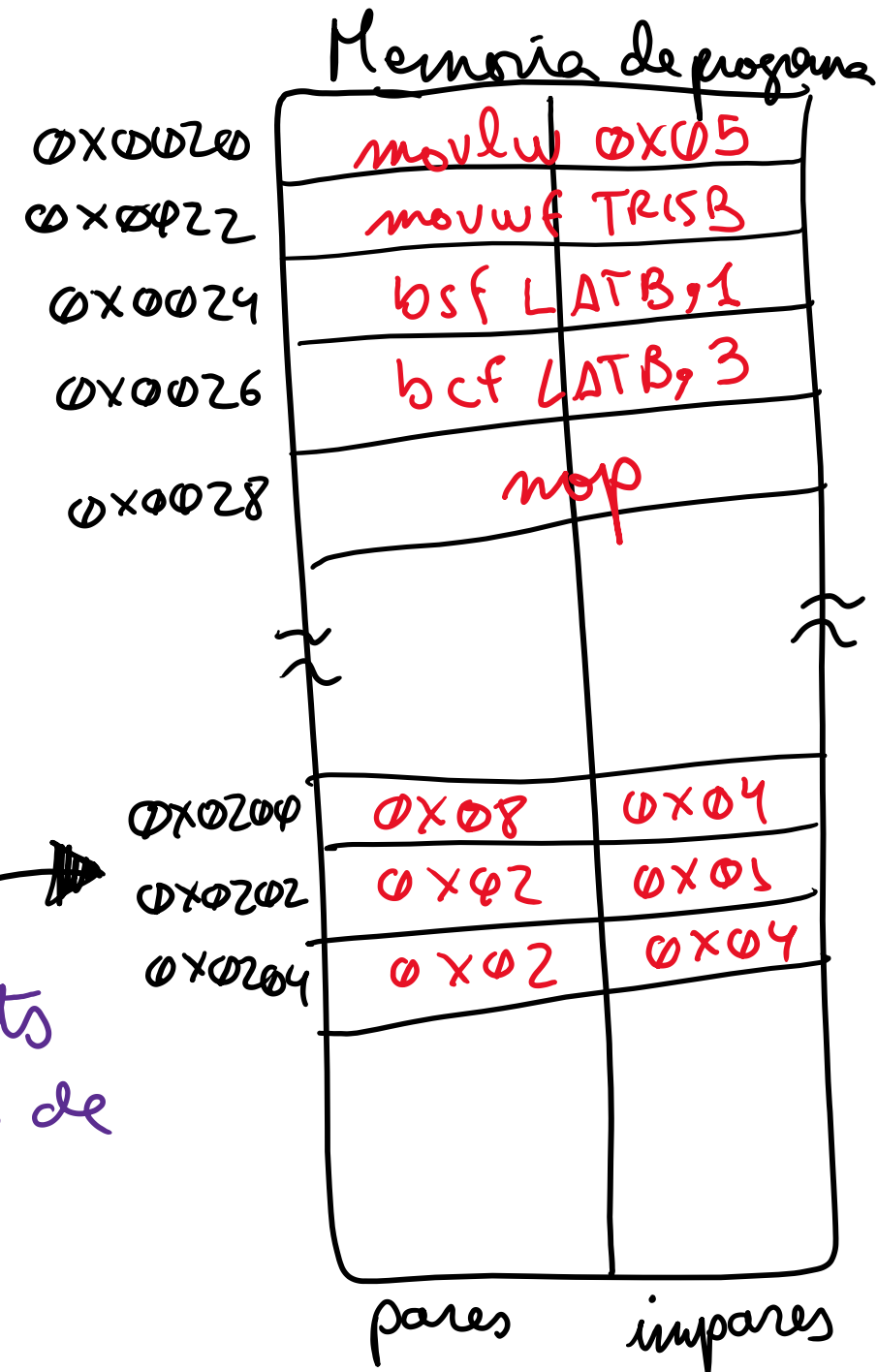
```
1      list p=18f4550          ;Modelo del microcontrolador
2      #include <p18f4550.inc>  ;librería de nombres
3
4      ;Zona de los bits de configuración del microcontrolador
5      CONFIG FOSC = XT_XT      ; Oscillator Selection bits (XT oscillator (XT))
6      CONFIG PWRT = ON         ; Power-up Timer Enable bit (PWRT enabled)
7      CONFIG BOR = OFF         ; Brown-out Reset Enable bits (Brown-out Reset disabled in hardware)
8      CONFIG WDT = OFF         ; Watchdog Timer Enable bit (WDT disabled (control is placed on user))
9      CONFIG PBADEN = OFF      ; PORTB A/D Enable bit (PORTB<4:0> pins are configured as digital I/O)
10     CONFIG LVP = OFF         ; Single-Supply ICSP Enable bit (Single-Supply ICSP disabled)
11
12     cblock 0x0020             ;Zona de declaración de etiquetas a los
13         cta_a                 ;registros GPR (variables)
14         cta_b
15         cta_c
16     endc
17
18     org 0x0200                ;En esta porción del programa colocaremos los datos
19     tablaton db 0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80, 0x40, 0x20, 0x10, 0x08, 0x04, 0x02
20
21     org 0x0000                ;Vector de reset
22     goto configura
23
24     org 0x0020                ;Zona del programa de usuario
25     configura:
26         clrf TRISD             ;Todo el puertoD como salida
27         movlw UPPER tablaton
28         movwf TBLPTRU
29         movlw HIGH tablaton
30         movwf TBLPTRH
31         movlw LOW tablaton
32         movwf TBLPTRL
```

# Ejercicio: Auto Fantástico KITT

```
34 inicio:
35     TBLRD*+           ;Lectura de lo que apunta e incrementa su posicion
36     movff TABLAT, LATD ;Mueve el contenido de TABLAT para LATD
37     call delaymon     ;Subrutina de retardo
38     movlw .13         ;Cargo 13 a W
39     cpgsft TBLPTRL     ;Pregunto si TBLPTR es mayor que W (13)
40     goto inicio       ;Cuando es falso
41     clrf TBLPTRL       ;Cuando es verdadero
42     goto inicio
```

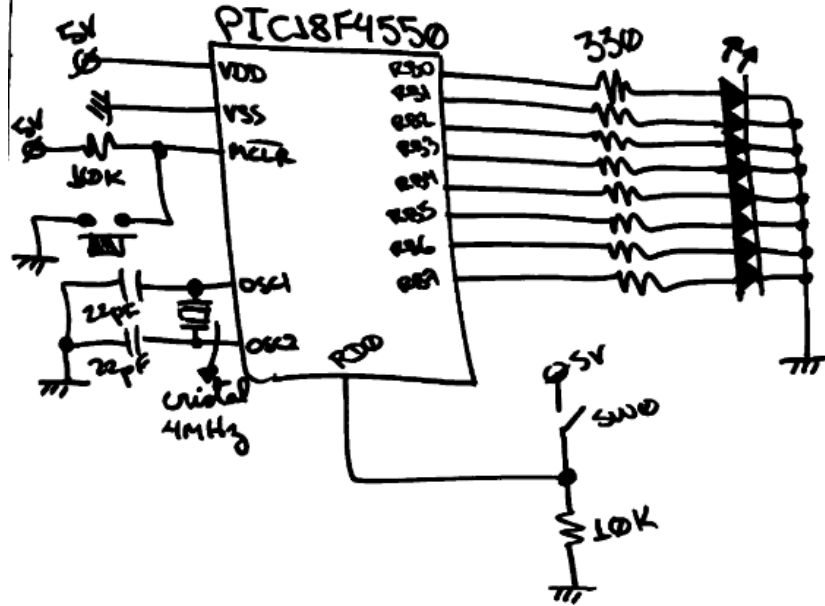
Nota: La rutina de retardo es la misma del ejemplo anterior

- Los datos almacenados en la memoria de programa serán de 8 bits en 8 bits, es decir, aprovechará el uso de direcciones pares e impares.

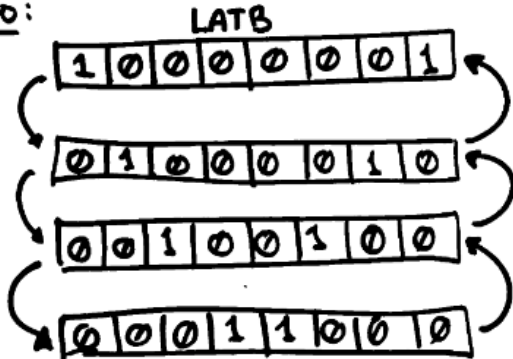


# Ejercicio: Auto Fantástico KITTbadguy

Ejercicio:



Funcionamiento:



Asignación: Hacer el diagrama de flujo

clr f [reg]

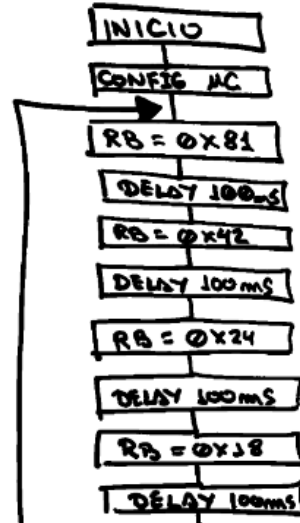
- ↳ coloca todos los bits de [reg] en cero
- set f [reg]
- ↳ coloca todos los bits de [reg] en uno

nomenclatura numérica

hex: 0x81

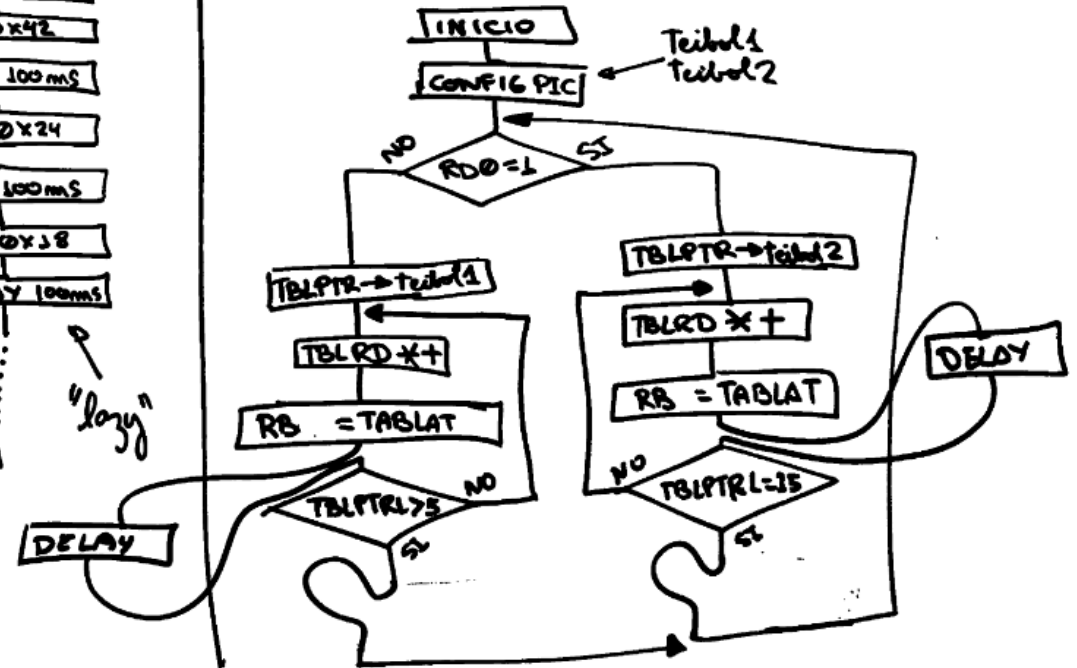
dec: 150, d'150'

bin: b'10100001



aplicando el "paguero"

Ahora con el botón:





# Ejercicio: Auto Fantástico KITTbadguy

```
19      org 0x0200
20      teibol1 db 0x81, 0x42, 0x24, 0x18, 0x24, 0x42
21
22      org 0x0300
23      teibol2 db 0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80, 0x40, 0x20, 0x10, 0x08, 0x04, 0x02
24
25      org 0x0000                ;Vector de reset
26      goto configura
27
28      org 0x0020                .
29      configura:
30      clrf TRISB                ;Todo el puertoB como salida
31
32      inicio:
33      btfss PORTD, 0
34      goto falsazo
35      verdaderazo:
36      movlw UPPER teibol1
37      movwf TBLPTRU
38      movlw HIGH teibol1
39      movwf TBLPTRH
40      movlw LOW teibol1
41      movwf TBLPTRL
42      otrazo1:
43      TBLRD*+
44      movff TABLAT, LATB
45      call delaymon
46      movlw .5
47      cpfsqt TBLPTRL
48      goto otrazo1
49      clrf TBLPTRL
50      goto inicio
```

```
52      falsazo:
53      movlw UPPER teibol2
54      movwf TBLPTRU
55      movlw HIGH teibol2
56      movwf TBLPTRH
57      movlw LOW teibol2
58      movwf TBLPTRL
59      otrazo2:
60      TBLRD*+
61      movff TABLAT, LATB
62      call delaymon
63      movlw .13
64      cpfsqt TBLPTRL
65      goto otrazo2
66      clrf TBLPTRL
67      goto inicio
68
```

Nota: Obviado:

- librerías
- bito de configuración
- variables de la subrutina de retardo.
- subrutina de retardo
- end.