```
1   ;-------------------------------------------------------------------------------;
2   ; Dal Degan Santiago - 45421137
3   ; Ejercicio 1 - Termotanque
4   ; En este programa controla un termotanque utilizando el PIC16F628A
5   ;-------------------------------------------------------------------------------;
6
7   ; se configura el pic
8   #include <p16f628a.inc>
9       LIST    P=16f628a
10
11      org 0
12
13  ; declaracion de variables
14
15  tmax    equ d'45'
16  maxagua equ d'110'
17  minagua equ d'50'
18  tagua   equ 0x20
19  cagua   equ 0x21
20  bomba   equ 0
21  bled    equ 1
22  tled    equ 2
23  resis   equ 3
24  resled  equ 4
25  maxled  equ 5
26  canilla equ 6
27  ; 0x20-0x21 utilizado
28
29  ;-----------SALIDAS-----------
30      bsf STATUS, 5
31      clrf TRISB ; Configuramos los pines B como salida
32      bcf STATUS, 5
33  ;-----------SALIDAS-----------
34
35  ;-----------VARIABLES PRUEBA-----------
36      movlw d'30'
37      movwf tagua
38      movlw d'100'
39      movwf cagua
40      movlw 0x00
41      movwf PORTB ; Limpiamos las salias
42  ;-----------VARIABLES PRUEBA-----------
43
44  ;-----------INICIO CICLO-----------
45  INICIO
46
47      clrwdt ; Limpiamos el watchdog
48      movlw maxagua ; Movemos el nivel maximo de agua a w
49      subwf cagua, 0 ; Restamos 110 - cantidad de agua
50
51      ; if
52      btfss STATUS, 2 ; Checkeamos si el bit de STATUS en la pos 2 es 0
53
54      ; false
55      goto PRENDER_BOMBA ; si el bit es 0 (es decir no hay suficiente agua), prendemos la bomba
56
57      ; true
58      goto CHECK_TEMP
59  ;-----------INICIO CICLO-----------
60
61  ;-----------PRENDER BOMBA-----------
62  PRENDER_BOMBA
63      bsf PORTB, bomba ; Prendemos la bomba
64      bsf PORTB, bled ; Prendemos el led marcador de la bomba
65
66  PRENDER_BOMBA_LOOP
67      incf cagua, 1 ; Simulamos el agua subiendo
68      movlw maxagua
69      subwf cagua, 0
70
71      ; if
```

```
72      btfss STATUS, 2 ; checkeamos nuevamente el nivel del agua
73
74      ; false
75      goto PRENDER_BOMBA_LOOP ; Si sigue bajo repetimos el loop
76
77      ; true
78      bcf PORTB, bomba ; Apagamos la bomba si el agua llego al nivel correcto
79      bcf PORTB, bled ; Apagamos el led de la bomba
80      bsf PORTB, maxled ; Prendemos el led de termotanque lleno
81      goto CHECK_TEMP
82  ;-----------PRENDER BOMBA-----------
83
84  ;-----------CHECK TEMPERATURA-----------
85  CHECK_TEMP
86      movlw tmax
87      subwf tagua, 0
88
89      ; if
90      btfss STATUS, 2 ; Checkeamos si la temperatura del agua es la maxima
91
92      ; false
93      goto PRENDER_RES ; Si no lo es prendemos la resistencia
94
95      ; true
96      bsf PORTB, tled ; Prendemos el led de temperatuar alcanzada
97      goto ABRIR_CANILLA ; Si lo es abrimos la canilla
98  ;-----------CHECK TEMPERATURA-----------
99
100 ;-----------PRENDER RESISTENCIA-----------
101 PRENDER_RES
102     bsf PORTB, resis
103
104 PRENDER_RES_LOOP
105     incf tagua, 1 ; Simulamos el aumento de temperatura del agua
106     movlw tmax
107     subwf tagua, 0
108
109     call DELAY250 ; Hacemos titilar el led
110     bsf PORTB, resled
111     call DELAY250
112     bcf PORTB, resled
113
114     ; if
115     btfss STATUS, 2 ; Checkeamos la temperatura del agua
116
117     ; false
118     goto PRENDER_RES_LOOP ; Si no se alcanzo la temperatura repetimos
119
120     ; true
121     ; Si la temperatura fue alcanzada apagamos la resistencia y el led
122     bcf PORTB, resis
123     bcf PORTB, resled
124     bsf PORTB, tled ; Prendemos el led de temperatuar alcanzada
125     goto ABRIR_CANILLA
126 ;-----------PRENDER RESISTENCIA-----------
127
128 ;-----------ABRIR CANILLA-----------
129 ABRIR_CANILLA
130     call DELAY1 ; Delay de 1 segundo antes de abrir la canilla
131
132     bsf PORTB, canilla ; Abrimos la canilla
133
134 ABRIR_CANILLA_LOOP
135     decf cagua, 1
136     movlw minagua
137     subwf cagua, 0
138
139     btfss STATUS, 2 ; Checkeamos si el agua llego a los 50 litros
140     ;false
141     goto ABRIR_CANILLA_LOOP
142
```

```
143      ;true
144      call DELAY1
145      call DELAY1
146      bcf PORTB, canilla
147      bcf PORTB, tled
148      bcf PORTB, maxled
149      goto INICIO
150  ;-----------ABRIR CANILLA-----------
151
152  ;-----------DELAY 1s-----------
153  DELAY1
154      ; estamos andando a 4Mhz
155      ; un ciclo de instruccion son 4 ciclos de relojs es decir 4/4 = 1Mhz
156      ; para calcular el tiempo hacemos 1/1Mhz = 1us
157      ; si queremos lograr un delay de 1s necesitamos
158      ; 1M ciclos de maquina
159      ; sin embargo como toma 3 ciclos de maquina hacer el proceso
160      ; dividimos 1M/3 = 333.333,33...
161      ; ya que no entra eso en un registro lo separaremos en 3
162      ; por cada valor del un registro el otro registro contara
163      ; regresivamente su valor
164      ; es decir reg1=10 reg=20, por cada 10 ciclos restando reg1
165      ; se restara uno de reg2
166      ; para saber los valores necesitamos reg1*reg2*reg3 = 333.333
167      ; raiz cubica 333.333 = 69.3
168
169      movlw d'69'
170      movwf 0x24
171  REG2
172      movlw d'69'
173      movwf 0x25
174  REG3
175      movlw d'70' ; ya que da un valor con coma a la 3ra le sumo uno
176      movwf 0x26 ; no es un delay exacto asi que no deberia importar
177
178  START
179      decfsz 0x26, 1
180      goto START
181      decfsz 0x25, 1
182      goto REG3
183      decfsz 0x24, 1
184      goto REG2
185      clrwdt
186      return
187  ;-----------DELAY 1s-----------
188
189  ;-----------DELAY 250ms-----------
190  DELAY250
191      ; La logica es la misma pero para 250ms
192
193      movlw d'43'
194      movwf 0x24
195  REG5
196      movlw d'43'
197      movwf 0x25
198  REG4
199      movlw d'44'
200      movwf 0x26
201
202  START1
203      decfsz 0x26, 1
204      goto START
205      decfsz 0x25, 1
206      goto REG5
207      decfsz 0x24, 1
208      goto REG4
209      clrwdt
210      return
211  ;-----------DELAY 500ms-----------
212      end
```