

## 1 Introduccion

Les propongo un reto constructivo, Mostremos las diferentes maneras de hacer un programa, asi veremos los estilos de programacion y la sintaxis de diferentes lenguajes de programacion y nos realimentaremos entre todos, no se preocupen empezamos desde cero!! y sobre todo NADIE TIENE LA RAZON, al menos en terminos de funcionalidad !! ( procuren poner una linea indicando la salida del compilador (nombre del compilador) para ver el porcentaje de memoria/codigo usado ) Un ejemplo cada semana asi dejamos 5 dias un tema y luego sera compilado en un pdf y subido al grupo

## 2 Enunciado

"Las n formas de prender y apagar un led" fout=1Hz. microcontrolador a usar: 16F1824. (si usas otra cosa, especificalo en tu codigo) puedes usar un retardo del compilador si quieres.

## 3 Codigos de ejemplo

### 3.1 XC8

---

```

1  /***** Clignotement de Leds *****/
2  micro= 16F1824 langage= C. compilateur= XC8
3  auteur= Julio A. Aguilar A. (Peru) - 2014 - Toulon - France
4  *****/
5  #include <xc.h>
6  #include <pic16f1824.h>
7  #pragma config CP=OFF, CPD=OFF, BOREN=OFF, WDTE=ON, PLEN=ON
8  #pragma config PWRTE=ON, FOSC=INTOSC, MCLRE=OFF, IESO=OFF, CLKOUTEN=OFF
9  #define _XTAL_FREQ 1000000
10 void config_hw(void);
11
12 void main(void){
13     config_hw();
14     while(1){
15         LATA ^= 0x01; // heartbeat
16         __delay_ms(500);
17     }
18
19 void config_hw(void){
20     TRISA = 0X00;
21     OSCCON = 0b01101000 ; // intosc -> 1Mhz
22     ANSELA = 0;
23     LATA=0;
24 }

```

---

### estadisticas del compilador

---

```

*****
Logfile:
*****
Microchip MPLAB XC8 C Compiler V1.32-Part Support Version:1.32(A)-Copyright(C)2014 Microchip Technology Inc.
Memory Summary:
Program space used 1Eh (30) of 1000h words ( 0.7%)
Data space used 4h ( 4) of 100h bytes ( 1.6%)
EEPROM space used 0h ( 0) of 100h bytes ( 0.0%)
Data stack space used 0h ( 0) of F0h bytes ( 0.0%)
Configuration bits used 2h ( 2) of 2h words (100.0%)
ID Location space used 0h ( 0) of 4h bytes ( 0.0%)

```

---

## 3.2 Proton Compiler

---

```

1  ***Led off on                ;***Pic =16f824
2  ;***Lenguaje = basic ;***Copilador = ;***proton ide
3  ;***Autor = ;Alexander Florez. ;***Bogota colombia ;- 2014
4
5  Device =16f824
6  Xtal = 4
7  Trisb = %00000000
8  Portb = %00000000
9  Symbol led = Portb.0
10 Loop:
11         Led = 1
12         Delays 200
13         Led = 0
14         Delays 200
15         Goto loop
16 ;End

```

---

## 3.3 AVR Bascom

nota del autor: Yo uso avr's, y tome uno de características similares, el attiny24, cristal de 20mHz, 2kb, 14 pines.

---

```

1  *****
2  micro= Attiny24 lenguaje= basic compilador= Bascom 2.7.5
3  *****
4  $regfile = "attiny24.dat"
5  $crystal = 1000000
6  Ddrb.0 = 1 : Portb.0 = 0
7  Do
8  Toggle Portb.0
9  Waitms 500
10 Loop
11 End

```

---

## estadísticas del compilador

---

```

Compiler : BASCOM-AVR LIBRARY V 2.0.7.5
Processor : TINY24
SRAM : 80 hex
EEPROM : 80 hex
ROMSIZE : 800 hex
FLASH USED : 7 %
BAUD : 9600 Baud
XTAL : 1000000 Hz

```

---

## 3.4 microchip assembler

---

```

1  ;*****
2  ; Filename: P16_LedBlinkAsm.asm *
3  ; Date: 04/08/2014 *
4  ; File Version: V1 *
5  ; Author: Ing. Jos Manuel Alvarado *
6  ; Company: ELECTRONICA Y ROBOTICA *
7  ;*****
8  ; Notes: *
9  ; Cambia el led de estado utilizando una interrupcion de 0.5s *

```

```
10 ;*****
11 list p=16f1824 ; directiva de lista para este microcontrolador
12 #include <p16f1824.inc> ; definicion de registros para el pic16f
13 ;Palabras de configuracion para este ejemplo
14
15 __CONFIG _CONFIG1, _FOSC_INTOSC&_WDTE_OFF&_PWRTE_ON&_BOREN_ON&_FCMEN_OFF&_IESO_OFF&_MCLRE_OFF&_CLKOUTEN_OFF
16 __CONFIG _CONFIG2, _LVP_OFF
17 ;*****
18 ORG 0x000 ; vector de reset
19 goto main ; va al inicio del programa
20 ;Vector de interrupcion de la familia PIC16XXX
21 ORG 0x004
22 ;Rutina de interrupcion
23 ;Cambio de estado en el pin RA0
24 banksel LATA
25 movlw 0x01
26 xorwf LATA,F
27 ;Vuelve a cargar contador de timer1 para desbordamiento a 0.5S, el periodo de cambio de estado es 1s
28 banksel TMR1
29 movlw 0xF7
30 movwf TMR1L
31 movlw 0xC2
32 movwf TMR1H
33 bcf PIR1,TMR1IF
34 ;termina interrupcion
35 retfie ; regresa de interrupcion
36 main
37 ;preparara portA para usar como led el pin RA0
38 banksel PORTA
39 clrf PORTA
40 banksel LATA
41 clrf LATA
42 banksel TRISA
43 clrf TRISA
44 banksel ANSELA
45 clrf ANSELA
46 ;configuracion del oscilador interno a 1MHz
47 banksel OSCCON
48 movlw 0x58
49 movwf OSCCON
50 ;El timer 1 funciona con un preescaler de 8, utiliza como fuente de reloj, el reloj de instrucciones
51 banksel TMR1
52 movlw 0x31
53 movwf T1CON
54 movlw 0xF7
55 movwf TMR1L
56 movlw 0xC2
57 movwf TMR1H
58 bcf PIR1,TMR1IF
59 banksel PIE1
60 bsf PIE1,TMR1IE;habilita interrupciones
61 movlw 0xC0
62 movwf INTCON
63 main_loop
64 ;puede poner otro codigo aqui
65 ;el codigo que ejecute en esta parte de programa
66 ;solo sera interrumpido por el desbordamiento del timer1
67 goto main_loop
68 END
```

## estadísticas del compilador

---

```
*****
Logfile:
*****
Microchip MPASM V5.57
Copyright (C) 2014 Microchip Technology Inc.

Memory Summary:
Program space used 27h (39) of 1000h words ( 0.9%)
Data space used 0h ( 0) of 100h bytes ( 0%)
EEPROM space used 0h ( 0) of 100h bytes ( 0.0%)
Data stack space used 0h ( 0) of F0h bytes ( 0.0%)
Configuration bits used 2h ( 2) of 2h words (100.0%)
ID Location space used 0h ( 0) of 4h bytes ( 0.0%)
```

---

## 3.5 Proton Compiler

---

```
1
2  '*****
3  '* Name : Proton IDE *
4  '* Author : Jacobo Sierra Arellanes, Oaxaca, Mxico *
5  '* Notice : Copyright (c) 2014 [select VIEW...EDITOR OPTIONS] *
6  '* : All Rights Reserved *
7  '* Date : 04/08/2014 *
8  '* Canal : https://www.youtube.com/user/siaj881108hocrrc05/videos *
9  Device = 16F1824
10 Xtal 4
11 TRISA=0
12 PORTA=0
13 Inicio:
14 PORTA=1
15 DelayMS 1000
16 PORTA=0
17 DelayMS 1000
18 GoTo Inicio
19 End
```

---

## estadísticas del compilador

---

```
'*****
'* Notes : 4 variable bytes used from possible 240 (1.67%) *
'* : 61 program bytes used from possible 4096 (1,49%) *
'*****
```

---

## 3.6 XC8

---

```
1
2  /*****
3  * File: Blink LED
4  * Author: Pedro Snchez (MrChunckuee) Electronica&Robotica
5  * Blog: http://mrchunckuee.blogspot.com/
6  * Email: mrchunckuee.psr@gmail.com
7  * Notas: Cambia es estado de LED_STATUS cada 0.5 segundos
8  * Fosc=4MHz, RA0=LED_STATUS, MCRL = +VCC
9  * IDE: MPLAB X, Compilador: XC8
```

```

10 * Created on 4 de agosto de 2014, 09:27 PM
11 *****/
12
13 #include <xc.h>
14
15 //Configuracion de FUSES del PIC16F1824
16 #pragma config CPD=OFF
17 #pragma config BOREN=ON
18 #pragma config IESO=OFF
19 #pragma config FOSC=INTOSC
20 #pragma config FCMEN=OFF
21 #pragma config MCLRE=ON
22 #pragma config WDTE=OFF
23 #pragma config CP=OFF
24 #pragma config PWRTE=ON
25 #pragma config CLKOUTEN=OFF
26 #pragma config PLLEN=OFF
27 #pragma config WRT=OFF
28 #pragma config STVREN=OFF
29 #pragma config LVP=OFF
30
31 #define _XTAL_FREQ 4000000
32 #define LED_STATUS LATA0
33
34 //Funciones utilizadas
35 void MCU_Init(void);
36
37 void main(void){
38 MCU_Init();
39 while(1){
40 LED_STATUS ^= 1;
41 __delay_ms(500);
42 }
43 }
44
45 void MCU_Init(void){
46 OSCCONbits.IRCF3=1;//
47 OSCCONbits.IRCF2=1;//
48 OSCCONbits.IRCF1=0;//
49 OSCCONbits.IRCF0=1;//Oscilador interno a 4MHz
50 ANSELA=0x00;//PORTA como digital
51 TRISA=0x00;//RA0 como salida
52 LED_STATUS=0;//Limpio RA0 antes de cargar un valor
53 }

```

---

## estadísticas del compilador

```

/*****/
Memory Summary:
Program space used 27h ( 39) of 1000h words ( 1.0%)
Data space used 5h ( 5) of 100h bytes ( 2.0%)
EEPROM space used 0h ( 0) of 100h bytes ( 0.0%)
Data stack space used 0h ( 0) of F0h bytes ( 0.0%)
Configuration bits used 2h ( 2) of 2h words (100.0%)
ID Location space used 0h ( 0) of 4h bytes ( 0.0%)

```

---

## 3.7 Pic CCS Compiler

---

```

1 /*****/

```

```

2 Nombre: Salvador H. Vallejo
3 Lugar: Oaxaca, Mxico
4 Fecha: 04.Ago.2014
5 ****/
6 #include <16f1824.h>
7 #fuses hs,nowdt,nomclr
8 #use delay(clock=20M)
9
10 /* PUERTOS */
11 #byte PortA= 0x0C
12 #byte PortC= 0x0E
13 #bit led= 0x0C.0
14
15 /* PROGRAMA PRINCIPAL */
16 void main (void){
17 set_tris_a(0b00000000);
18 set_tris_c(0b00000000);
19
20 portA=0;
21 portC=0;
22
23 do{
24 led=0;
25 delay_ms(500);
26 led=1;
27 delay_ms(500);
28
29 }
30 while(true);
31 }

```

---

## estadísticas del compilador

```

*****
Compilador CCS v4.140
ROM:
58(1%)
58(1%) including unused fragments
0 Average locations per line
6 Average locations per statement
RAM:
5(2%) at main () level
17(7%) worst case
*****

```

---

## 3.8 microchip assembler

```

1 ;*****
2 ; Filename: P16_LedBlinkAsm2.asm *
3 ; Date: 05/08/2014 *
4 ; File Version: V2 *
5 ; Author: Ing. Jos Manuel Alvarado *
6 ; Company: ELECTRONICA Y ROBOTICA *
7 ;*****
8 ; Notes: *
9 ; Cambia el led de estado utilizando retardos por software *
10 ;*****
11 list p=16f1824 ; directiva de lista para este microcontrolador
12 #include <p16f1824.inc> ; definicion de registros para el pic16f

```

```

13 ;Palabras de configuracion para este ejemplo
14 __CONFIG _CONFIG1, _FOSC_INTOSC&_WDTE_OFF&_PWRTE_ON&_BOREN_ON&_FCMEN_OFF&_IESO_OFF&_MCLRE_OFF&_CLKOUTEN_OFF
15 __CONFIG _CONFIG2, _LVP_OFF
16 ;*****
17 ;Bloque de variables en Ram
18 CBLOCK 0x20
19 CounterB
20 CounterA
21 ENDC
22 ;*****
23 ORG 0x000 ; vector de reset
24 goto main ; va al inicio del programa
25 ;Vector de interrupcion de la familia PIC16XXX
26 ORG 0x004
27 retfie ; regresa de interrupcion
28 main
29 ;preparara portA para usar como led el pin RA0
30 banksel TRISA
31 clrf TRISA
32 ;configuracion del oscilador interno a 1MHz
33 movlw 0x58
34 movwf OSCCON
35 banksel ANSELA
36 clrf ANSELA
37 ;Cambio de estado de RA0 utilizando un retardo por software
38 banksel LATA
39 main_loop
40 movlw 0x01
41 xorwf LATA,F
42 call delay_half_second
43 goto main_loop
44 ;Retardo generado utilizando el programa PICLOOPS
45 delay_half_second
46 movlw D'163'
47 movwf CounterB
48 movlw D'84'
49 movwf CounterA
50 loop decfsz CounterA,1
51 goto loop
52 decfsz CounterB,1
53 goto loop
54 retlw 0
55
56 END

```

---

## estadísticas del compilador

```

*****
Logfile:
*****
Microchip MPASM V5.57
Copyright (C) 2014 Microchip Technology Inc.
Memory Summary:
Program space used 27h (26) of 1000h words ( 0.6%)
Data space used 0h ( 2) of 100h bytes ( 0.78%)
EEPROM space used 0h ( 0) of 100h bytes ( 0.0%)
Data stack space used 0h ( 0) of F0h bytes ( 0.0%)
Configuration bits used 2h ( 2) of 2h words (100.0%)
ID Location space used 0h ( 0) of 4h bytes ( 0.0%)

```

---