

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

PROPUESTA 2: Jugando (otra vez) con leds

microcontrolador a usar: 12F675. (si usas otra cosa, especificalo en tu codigo) Objetivo: controlar, mediante un registro de desplazamiento, 74595 y mostrar una secuencia de leds (para standarizar la secuencia : empieza en 10000001 y termina en 00011000 y se repite de nuevo, como las del coche fantastico) la velocidad del efecto debe controlarse con un potenciómetro.

3 Codigos de ejemplo

3.1 XC8

```

1  /*
2  * Exemple de pilotage du registra dcalage 74HC595 et Adc
3  * Author: Julio Alexander Aguilar Angulo (juloaguilar@gmail.com)
4  * ISEN - Toulon: http://www.isen.fr/toulon/
5  * compilateur: XC8. mcu: 12F1840. Intosc:4Mhz
6  * */
7  #include <xc.h>
8  #include <pic12f1840.h>
9  #pragma config CP=OFF, CPD=OFF, BOREN=OFF, WDTE=OFF
10 #pragma config PWRTE=OFF, FOSC=INTOSC, MCLRE=OFF, PLLEN=OFF, LVP=OFF
11 /* Declaration des surnoms des ports
12 */
13 #define SpiData LATAbits.LATA4
14 #define SpiClock LATAbits.LATA1
15 #define ChipLatch LATAbits.LATA5
16 #define ChipSelect LATAbits.LATA2
17 /*Tableau de patrons afficher*/
18 char anypattern[]={0b10000001,0b01000010,0b00100100,0b00011000,0b00011000,0b00100100,0b01000010,0b10000001};
19 char index, j,k;
20 /* ISR: affichage de chaque patron lors du dbordement du timer */
21 void interrupt global_isr(void)
22 {
23     if(PIR1bits.TMR1IF==1)
24     {
25         j++;
26         if (j==ADRESH ) {
27             ChipSelect=1;
28             SSP1BUF=anypattern[index];
29             while(PIR1bits.SSP1IF==0){}
30             PIR1bits.SSP1IF=0;
31             ChipLatch=1;
32             ChipLatch=0;
33             ChipSelect=0;
34             index++;
35             if (index == 8) index=0;
36             j=0;
37         }
38         ADCON0=0x01; //RA0

```

```

39  ADCON0bits.ADGO=1; // lancement de la conversion
40  while (ADCON0bits.ADGO) {} // attente
41  TMR1H=0xFF;TMR1L=0x00;
42  PIR1bits.TMR1IF=1; // Enable interrupt on PR1=TMR1
43
44  };
45
46  }
47  /*Programme principal: Lecture en boucle du convertisseur analogique-numrique*/
48  int main() {
49  OSCCON=0b01001000; //4Mhz internal osc
50  TRISA=0b00000001; // RA0 pot, le reste sorties
51  ANSELA=0; // desactive analog
52  CM1CON0=0; // desactiver comparateurs
53  CM1CON1=0; //
54  APFCON=0;
55  ANSELAbits.ANSA0=1; // config RA0 analog
56  TRISA=0b00000001; // config I/O
57  APFCONbits.SDOSEL=1; // DOut sur RA4
58  SSP1STAT=0b01000000; // spi setup
59  SSP1CON1=0b00100000; // F=Fosc/4
60  TMR1H = 0xFF; // MSB du tim1
61  TMR1L = 0; // LSB du tim1
62  T1CON=0b00111001; // config et lance le timer1
63  /* configuration des interruptions */
64  INTCONbits.PEIE=0;
65  INTCONbits.GIE=0;
66  OPTION_REG=0; // concerne Timer0 et RA2 int
67  PIE2=0;
68  PIE1bits.TMR1IE=1;
69  INTCON=0;
70  PIR1=0;
71  index=0;
72  INTCONbits.PEIE=1;
73  INTCONbits.GIE=1;
74  while(1)
75  {
76  }
77
78  }

```

estadísticas 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 7Bh ( 123) of 1000h words ( 3.0%)
Data space used Ch ( 12) of 100h bytes ( 4.7%)
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 AVR Bascom

```

1  '*****
2  'author: Juan Litardo

```

```

3 'Micro : Attiny13
4 'Memoria : 1Kb
5 'Cristal : 9.6MHz
6 'Bascom Avr - Basic
7 '*****
8 $regfile = "attiny13.dat"
9 $crystal = 9600000
10 $hwstack = 0
11
12 Ddrb = &B1011 : Portb = &B0100
13
14 Config Adc = Single , Prescaler = Auto , Reference = Internal
15 Start Adc
16 Dim Tempo As Word , Var As Byte , A As Byte
17
18 Do
19 For A = 0 To 31
20 Portb.3 = 0
21 Tempo = Getadc(1)
22 Portb.0 = 1
23 Var = Lookup(a , Tabla)
24 Portb.1 = Var
25 Waitms 10
26 Portb.0 = 0
27 If A = 0 Or A = 8 Or A = 16 Or A = 24 Then
28 Portb.3 = 1
29 Tempo = Tempo / 4
30 Waitms Tempo
31 End If
32 Next
33 Loop
34 End
35
36 Tabla:
37 Data 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 1
38 Data 0 , 1 , 0 , 0 , 0 , 0 , 0 , 1 , 0
39 Data 0 , 0 , 1 , 0 , 0 , 0 , 1 , 0 , 0
40 Data 0 , 0 , 0 , 1 , 1 , 0 , 0 , 0 , 0

```

estadísticas del compilador

```
'Flash Used : 52 %
```

3.3 Proton Compiler

```

1
2 '*****
3 '* Name : Desplazamiento.BAS *
4 '* Author : Jacobo AS *
5 '* Notice : Se utiliza un registro de desplazamiento 74LS164 *
6 '* : PIC12F675 *
7 '* Date : 18/08/2014, Oaxaca, Mexico *
8 '* Version : 1.0 *
9 '* Notes :https://www.youtube.com/user/siaj881108hocrrc05/videos... *
10 '*****
11
12 Device = 12F675
13
14 Reminders = OFF

```

```

15 Config CPD_OFF,CP_OFF,BODEN_OFF,MCLRE_OFF,PWRTE_OFF,WDT_OFF ,INTRC_OSC_NOCLKOUT
16 Reminders = On
17
18 Xtal = 4
19 Declare Adin_Res = 10
20 Declare Adin_Tad = FRC
21 Declare Adin_Stime = 50
22
23 CMCON = 7
24 TRISIO = %00000001
25 ANSEL = %00100001
26 ADCON0 = %10000011
27 GPIO = 0
28 Dim und As Byte
29 Dim u As Byte
30 Dim i As Byte
31 Dim Tiempo As Word
32
33
34 Symbol dato = GPIO.5
35 Symbol clock = GPIO.4
36
37 main:
38 Tiempo = ADIn 0
39 For i = 0 To 6
40 und = LookUp i,[129, 66, 36, 24, 36, 66, 129]
41 GoSub enviar
42 DelayMS Tiempo
43 Next i
44
45 GoTo main
46 End
47
48 enviar:
49 For u = 0 To 7
50 If und.7 = 0 Then
51 dato = 0
52 clock = 0
53 DelayMS 1
54 clock = 1
55 DelayMS 1
56 Else
57 dato = 1
58 clock = 0
59 DelayMS 1
60 clock = 1
61 DelayMS 1
62 EndIf
63
64 und = und<<1
65
66 Next u
67 Return

```

estadísticas del compilador

```

'*****
'* :Proton IDE::143 palabras usadas de 1024 (13,96%) *
'* 13 bytes usados de un posible de 64 (20,31%) *
'*****

```

3.4 XC8

```

1
2 * File: Registro de desplazamineto
3 * Author: Pedro Snchez (MrChunckuee) - Electronica&Robotica
4 * Blog: http://mrchunckuee.blogspot.com/
5 * Email: mrchunckuee.psr@gmail.com
6 * Notas: Lueces con efecto del auto fantatico usando resgistro
7 * de desplazamiento, la velocidad del efecto varia con el
8 * potenciometro de 10K conectado en AN0. PIC12F675, 74HC595
9 * Fosc=4MHz, AN0=POTENCIOMETRO 10K, MCRL = +VCC
10 * IDE: MPLAB X, Compilador: XC8
11 * Created on 11 de agosto de 2014, 11:22 PM
12 *****/
13
14 #include <xc.h>
15
16 //Configuracion de FUSES
17 #pragma config BOREN=OFF
18 #pragma config CPD=OFF
19 #pragma config FOSC=INTRCIO
20 #pragma config MCLRE=ON
21 #pragma config WDTE=OFF
22 #pragma config CP=OFF
23 #pragma config PWRT=ON
24
25 #define _XTAL_FREQ 4000000
26
27 #define ADC_POT GPIObits.GP0
28 #define SH_CLK GPIObits.GP1
29 #define PIN_DATA GPIObits.GP2
30 #define ST_CLK GPIObits.GP4
31
32 unsigned char LEDs_DATA[7]={0b10000001,
33 0b01000010,
34 0b00100100,
35 0b00011000,
36 0b00100100,
37 0b01000010,
38 0b10000001};
39
40 //Declaracion de funciones utilizadas
41 void MCU_Init(void);
42 void Send_Data(unsigned char DATA);
43 unsigned int Read_ADC(void);
44 void Delay(unsigned int i);
45
46 void main (void){
47     unsigned char i;
48     MCU_Init();
49     while(1){
50         for(i=0; i<6; i++){
51             Send_Data(LEDs_DATA[i]);
52             Delay(Read_ADC());
53         }
54     }
55 }
56
57 void MCU_Init(void){
58     //Configuracion del ADC
59     ADCONbits.ADFM = 1;
60     ADCONbits.VCFG = 0; //Voltage Reference VDD

```

```

61  ADCON0bits.CHS = 0b00; //Select chanel ANO
62  ANSELbits.ADCS = 0b100;
63  ANSELbits.ANS = 0b0001;
64  ADCON0bits.ADON = 1; //Turn ADC on
65
66  TRISIO=0b0000001; //Solo ANO como entrada lo demas es salida
67
68  //Limpiamos el puerto
69  ADC_POT=0;
70  SH_CLK=0;
71  PIN_DATA=0;
72  ST_CLK=0;
73  }
74
75  void Send_Data(unsigned char DATA){
76  unsigned char LOOP, FLAG;
77  for (LOOP=0; LOOP<8; LOOP++){
78  FLAG=DATA&0x01;
79  if(FLAG==0)
80  PIN_DATA=0;
81  else
82  PIN_DATA=1;
83  SH_CLK = 1;
84  SH_CLK = 0;
85  DATA >>= 1;
86  }
87  ST_CLK = 1;
88  ST_CLK = 0;
89  }
90
91  unsigned int Read_ADC(void){
92  int RESULT_ADC=0;
93  ADCON0bits.GO = 1;
94  while(ADCON0bits.GO);
95  RESULT_ADC = ADRESH << 8;
96  RESULT_ADC = RESULT_ADC + ADRESL;
97  return (RESULT_ADC);
98  }
99
100 void Delay(unsigned int i){
101 unsigned int j;
102 for(j=0;j<i;j++){
103 __delay_ms(1);
104 }
105 }

```

estadísticas del compilador

```

/*****/
Memory Summary:
Program space used B0h ( 176) of 400h words ( 17.2%)
Data space used 16h ( 22) of 40h bytes ( 34.4%)
EEPROM space used 0h ( 0) of 80h bytes ( 0.0%)
Data stack space used 0h ( 0) of 2Ah bytes ( 0.0%)
Configuration bits used 1h ( 1) of 1h word (100.0%)
ID Location space used 0h ( 0) of 4h bytes ( 0.0%)

```

3.5 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
*****
```
