'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* www.ideastechnology.com \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'

' \* Nombre proyecto:

' CONTROL DE RPM DE UN MOTOR TRIFASICO DE 1 HP

' \* Descripcion:

' \* Configuraciones:

' Microcontrolador: PIC16F886

' Oscillator: INT\_RC\_OSC\_NOCLKOUT, 4MHz

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\* Nombre proyecto \*\*\*\*\*

program Adquisicion

'\*\*\*\*\* Declaración de Variables \*\*\*\*\*

dim mtempin,mtempout,n,count,inicio as word

dim tempin,tempout,receive,flanco as byte

dim txt as string[2]

dim recivio as byte

dim velocidad,HZ as word

dim automatico as byte

dim salto as byte

dim operacion as byte

dim Frecuencia as word

dim alto\_adc, bajo\_adc as byte

dim adc\_pot as byte

dim velocidadhz as word

dim tiempo as float

dim vueltas as byte

'\*\*\*\*\* Programa principal main \*\*\*\*\*

'-----------------------------------------

sub procedure interrupt ' Interrupt service routine

if (UART1\_Data\_Ready() = 1) then

receive = UART1\_Read()

velocidad=receive

end if

if INTCON.INTF = 1 then ' External interupt (RB0 pin) ?

' intcon = intcon or %00100000

inc(vueltas)

' if operacion = 1 then

' tiempo = 0

' intcon = intcon and %11011111

' tiempo = count\*0.3251

' tiempo = tiempo + (255-tmr0)\*0.001275

' operacion = 0

' vueltas=0

' end if

' tmr0 = 0

' count = 0

Clearbit(INTCON,INTF)

Setbit(INTCON,INTE)

end if

' if intcon.TMR0IF = 1 then ' 32.51 ms

' operacion = 1

' count = inc(count)

' intcon.TMR0IF = 0

' tmr0 = 0

' end if

end sub

'-\*---------------------------------------

main:

OSCCON = 0X75

OPTION\_REG=%01000111

INTCON = %11000000 ' Enable external interrupts bit(7) and bit(6) de permiso q no se controlan con INTCON

PIE1 = %00100000

' Configuracion de Puertos analogicos / digitales AN0 - AN13

ANSEL = %00000001 'REGISTRO CONFIG. O DIGITAL, 1 ANALOGICO

ANSELH = %00000000 'REGISTRO CONFIG. O DIGITAL, 1 ANALOGICO

' Configuracion de Puertos como entradas / salidas

trisa = %00000001 ' Configuracion IN/OUT PORTA

trisb = %00000001 ' Configuracion IN/OUT PORTB

trisc = %10000000 ' Configuracion IN/OUT PORTC

' Reset de salidas

PORTA = %00000000 ' Inicializacion PORTA

PORTC = %00000000 ' Inicializacion PORTC

'USART

Uart1\_Init(9600) ' Initialize USART module

n=0

automatico=0

salto = 1

Frecuencia = 0

operacion =0

count = 0

tiempo = 0

vueltas=0

while true

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

' for n=0 to 64

' adc\_pot = adc\_pot + ADC\_read(0)

' next n

' adc\_pot = (adc\_pot/64)

' bajo\_adc= lo(velocidad)

velocidadhz = velocidad \* 113

PORTA = lo(velocidadhz)

alto\_adc= hi(velocidadhz)

PORTB.5 = alto\_adc.0

PORTB.4 = alto\_adc.1

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

vueltas = 0

INTCON.INTE=1

delay\_ms(200)

delay\_ms(200)

delay\_ms(200)

INTCON.INTE= 0

'PORTA = vueltas

UART1\_Write(0x66)

byteToStr(vueltas, txt)

UART1\_Write\_text(txt)

'adc\_pot = byte(vueltas)

'adc\_pot = lo(vueltas)

'byteToStr(vueltas, txt)

' UART1\_Write\_Text(txt)

' adc\_pot = lo(vueltas)

' UART1\_Write(adc\_pot)

' adc\_pot = hi(vueltas)

' UART1\_Write(adc\_pot)

' adc\_pot = higher(vueltas)

' UART1\_Write(adc\_pot)

' adc\_pot = highest(vueltas)

' UART1\_Write(adc\_pot)

' WordToStr(tiempo, txt)

' UART1\_Write\_text(txt)

' delay\_ms(100)

' porta.1 = porta.1 xor 1

adc\_pot = 0

wend

end.



