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

'

' \* Nombre proyecto:

' SISTEMA DE ADQUISICION Y ALMACENAMIENTO DE DATOS CON MATLAB

' \* Descripcion:

' Este proyecto consiste en leer un dato analogico proveniente de un potenciometro

' para controlar el nivel de PWM que le aplicamos al motor por medio del puente H

' la señal digital que proviene del encoder optico es procesada y tranformada

' a revoluciones por minuto (RPM) y posteriormente enviada por comunicacion serial

' RS232 al computador donde MATLAB se encargara de tomar los datos de (PWM y RPM).

' \* Configuraciones:

' Microcontrolador: PIC16F886

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

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

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

program Adquisicion

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

dim velocidad,prom,mldr1,mldr2,cont as word

dim LDR1,LDR2,n,receive as byte

dim txt as string[4]

dim recivio as byte

dim velocidad\_2 as word

dim automatico as byte

dim salto as byte

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

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

sub procedure interrupt ' Interrupt service routine

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

receive = UART1\_Read()

select case salto

case 1

if receive = 0xA0 then

salto =2

else

salto =1

end if

case 2

salto =3

velocidad= (receive\*254)/9

' portA=receive

recivio = 1

case 3

salto =1

automatico = receive ' automatico = 1 y no automatico = 0

end select

end if

end sub

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

main:

OPTION\_REG=%10000000

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

'T1CON = $30

'PIR1.TMR1IF = 0 ' clear TMR1IF

'TMR1H = $0B ' initialize Timer1 register

'TMR1L = $DB

PIE1 = %00100000 ' enable Timer1 interrupt

' Configuracion de Puertos analogicos / digitales AN0 - AN13

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

ANSELH = %00001010 'REGISTRO CONFIG. O DIGITAL, 1 ANALOGICO

' Configuracion de Puertos como entradas / salidas

trisa = %00000001 ' Configuracion IN/OUT PORTA

trisb = %00011000 ' 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

' Lazo infinito Bucle While

PWM1\_Init(1000) ' Initialize PWM1 module at 5KHz

PWM2\_Init(1000) ' Initialize PWM2 module at 5KHz

PWM1\_Start() ' start PWM1

PWM2\_Start() ' start PWM2

PWM1\_Set\_Duty(0)

PWM2\_Set\_Duty(0)

LDR1 = ADC\_Read(9)

LDR2 = ADC\_Read(11)

prom=(LDR1+LDR2)/2 + 10

n=0

automatico=0

salto = 1

while true

n=0

while (n<40)

if automatico = 0 then

velocidad=ADC\_Read(0)

end if

LDR1 = ADC\_Read(9)

delay\_us(5)

LDR2 = ADC\_Read(11)

delay\_us(5)

mldr1=mldr1+LDR1

mldr2=mldr2+LDR2

n=n+1

wend

if recivio= 1 then

PWM1\_Set\_Duty(velocidad)

PWM2\_Set\_Duty(velocidad)

recivio=0

end if

mldr1=mldr1/40

mldr2=mldr2/40

if (mldr1>prom)or (mldr2>prom) then

if (mldr1 > mldr2) then

portb.1=0

PWM1\_Set\_Duty(Velocidad)

portb.2=1

PWM2\_Set\_Duty(Velocidad)

else

portb.1=1

PWM1\_Set\_Duty(Velocidad)

portb.2=0

PWM2\_Set\_Duty(Velocidad)

end if

else

PWM1\_Set\_Duty(0)

PWM2\_Set\_Duty(0)

end if

inc(cont)

if (cont > 200) then

UART1\_Write\_text("s")

ByteToStr(mldr1, txt)

UART1\_Write\_text(txt)

UART1\_Write\_text("w")

ByteToStr(mldr2, txt)

UART1\_Write\_text(txt)

porta.1 = porta.1 xor 1

cont =0

end if

' ifk (UART1\_Data\_Ready() = 1) then

' receive = UART1\_Read()

'' if receive <= 100 then '' p - me envian modo pc

'' ''modo=1

'' Velocidad=receive

'' end if

' if receive= 109 then '' m - me envian modo manual

' analogico = Adc\_Read(0)>>2

' delay\_ms(5)

' Velocidad=analogico

' ''modo=2

' else

' Velocidad=(receive - 48)\*28.33

' end if

' end if

' if Velocidad >=250 then

' PWM1\_Stop()

' else

' PWM1\_Start()

' end if

'

' PWM1\_Set\_Duty(Velocidad)

' PORTA=255-Velocidad

' ByteToStr(Frecuencia,tx\_f)

' ByteToStr(255-Velocidad,tx\_v)

' UART1\_Write(116)''t

' UART1\_Write\_Text(tx\_v)

' UART1\_Write(102)''f

' UART1\_Write\_Text(tx\_f)

' UART1\_Write(10)

' UART1\_Write(13)

' delay\_ms(50)

'PWM2\_Set\_Duty(Velocidad)

wend

end.



