

**ESCUELA POLITECNICA NACIONAL**

**FACULTAD DE INGENIERIA ELECTRICA**

**TESIS PREVIA A LA OBTENCION DEL TITULO**

**DE INGENIERO EN**

**ELECTRONICA Y CONTROL**

**DISEÑO Y CONSTRUCCION DE UN PROTOTIPO**

**DE UN PROGRAMADOR PARA**

**MICROCONTROLADORES PIC16C5X**

**JUAN FRANCISCO SORIA AYALA**

**Quito, Enero de 1998**

***ANEXO 3***  
***LISTADOS DE LOS PROGRAMAS***  
***IMPLEMENTADOS***

## LISTADO DEL PROGRAMA PARA EL PROGRAMADOR

; PROGRAMA PARA PROGRAMAR MICROCONTROLADORES DE LA FAMILIA PIC16C5X  
; ASIGNACION DE ETIQUETAS

```

;/////////////////////////////////////////////////////////////////
;/  R7 SIRVE COMO EL BYTE MAS SIGNIFICATIVO DE LA DIRECCION           /
;/  R6 SIRVE PARA EL BYTE MENOS SIGNIFICATIVO DE LA DIRECCION       /
;/  R5 SE UTILIZA PARA GRABAR EL # DE DATOS QUE SE VA A PROGRAMAR   /
;/////////////////////////////////////////////////////////////////

```

```

    DSERIAL     EQU    7FH    ;DATO SERIAL
    COMANDO     EQU    7EH    ;COMANDO
    TAMAÑO      EQU    7DH    ;TAMAÑO DE LA MEMORIA
    BSERIAL     BIT    00H    ;BANDERA SERIAL
    MICRO       BIT    01H    ;TIPO DE MICRO
    RESETEO     BIT    02H    ;PERMITE INICIALIZAR EL PROGRAMADOR
;/////////////////////////////////////////////////////////////////

```

```

OR      0000H
      LJMPP INICIO ;PROGRAMA PRINCIPAL

```

```

ORG     0023H
      LJMPP SERIAL ;INTERRUPCION SERIAL

```

```

ORG     0030H
;
;/////////////////////////////////////////////////////////////////

```

```

INICIO:                                ;INICIALIZACION DEL SISTEMA
      MOV    SP,#2FH                    ;STACK POINTER DESDE (2FH)
      MOV    IE,#10010000B              ;HABILITO INTERRUPCION SERIAL
      MOV    PCON,#00H                  ;SMOD = 0
      MOV    TMOD,#00100000B           ;TIMER 1 EN MODO 2 PARA GENERAR EL BAUD RATE
      MOV    SCON,#01010000B           ;PROGRAMO EL PORTICO SERIAL
      MOV    TH1,#0FEH                  ;CON ESTOS VALORES SE OBTIENE UN BAUD
      MOV    TL1,#0FEH                  ;RATE DE 9600 CON UN CRISTAL DE 7,159MHz
      SETB   TR1                        ;ARRANCA TIMER 1 (BAUDE RATE)
      CLR    BSERIAL                    ;BORRO LA BANDERA SERIAL

```

; PONGO CONDICIONES INICIALES AL SOCKET DE PROGRAMACION

```

COMANDOS:
      LCALL  CONDICION

```

```

; DETERMINO SI EXISTE COMUNICACION CON EL COMPUTADOR
      JNB    BSERIAL,$                  ;ESPERO INTERRUPCION SERIAL (DATO)
      CLR    BSERIAL                    ;BORRO LA BANDERA SERIAL
      MOV    A,DSERIAL                  ;MUEVO AL ACC EL DATO SERIAL
      CJNE  A,#06H,COMANDO1             ;
      MOV    SBUF,A                      ;TRANSMITO EL DATO RECIVIDO
      LJMPP COMANDOS

```

```

; SI EXISTE COMUNICACION ESPERO LOS COMANDOS
      ;COMANDO 0XH PARA LECTURA DE LAS MEMORIAS DE LOS PICS
      ;COMANDO 8XH PARA LA PROGRAMACION DE LOS MICROS PICS

```

```

COMANDO1:
      LCALL  FORMATO
      MOV    SBUF,A
      MOV    A,DSERIAL

```

```

MOV    COMANDO,A
JNB    BSERIAL,$           ;ESPERO INTERRUPCION SERIAL (DATO)
CLR    BSERIAL             ;BORRO LA BANDERA SERIAL
MOV    A,DSERIAL
CJNE   A,#00H,COMANDOS
MOV    A,COMANDO
JB     ACC.7,PROGRAMACION
LJMP   LECTURA

```

```
;
```

```
////////////////////////////////////
```

```
PROGRAMACION:
```

```

*****
;*      SUBROUTINAS PARA PROGRAMAR LOS MICROCONTROLADORES PIC16C5X      *
*****
    LCALL SELECCION          ;SELECCIONO EL DISPOSITIVO
    JNB   MICRO,PROGPIC18   ;SELECCIONO QUE MICRO VOY A PROGRAMAR
    LJMP  PROGPIC28

```

```
PROGPIC18:
```

```

*****
;*      PROGRAMA LA CONFIGURACION DE LA PALABRA EN LOS PICS DE 18 PINES  *
*****
    CLR   P0.4
    SETB  P1.4             ;CONECTO VDD AL MICRO
    CLR   P1.5             ;PONGO EN BAJO OSC1
    SETB  P0.2             ;PONGO EN ALTO TOCK1
    MOV   A,COMANDO       ;ESCOJO SOLO LOS BITS DE CONFIGURACION

```

```
; DETERMINO SI HABILITO O NO EL CODIGO DE PROTECCION
```

```

    JNB   ACC.3,CODEP1    ;REVISO SI SE PROGRAMA EL CODIGO DE PROTECCION
    SETB  P0.1             ;RA3 ;RA3
    LJMP  FINCODE1

```

```
CODEP1: ;HABILITO EL CODIGO DE PROTECCION
```

```
    CLR   P0.1             ;RA3
```

```
FINCODE1:
```

```
; DETERMINO SI HABILITO O NO WATCHDOGTIMER
```

```

    MOV   A,COMANDO
    JB    ACC.4,WDTE1     ;CHEQUEO SI PROGRAMA EL WATCHDOGTIMER
    CLR   P0.0             ;DESABILITO WDTE
    LJMP  FINWDTE1

```

```
WDTE1:
```

```
    SETB  P0.0             ;HABILITO WDTE
```

```
FINWDTE1:
```

```

*****
;*      SETEO QUE TIPO DE OSCILADOR VOY A PROGRAMAR                       *
*****

```

```

    MOV   A,COMANDO
    ANL   A,#60H
    SWAP  A
    RR    A
    CJNE  A,#03H,OSCILA11
    SETB  P2.5             ;SETEO LOS BITS PARA CONFIGURAR EL OSCILADOR
    SETB  P2.6             ;EN RC (OSCILADOR RC)
    LJMP  FINOSCILA1

```

```
OSCILA11:
```

```

    CJNE  A,#02H,OSCILA12
    CLR   P2.5             ;SETEO LOS BITS PARA CONFIGURAR EL OSCILADOR

```

```

        SETB P2.6                ;EN HS (CRISTAL DE ALTA VELOCIDAD)
        LJM  FINOSCILA1
OSCILA12:
        CJNE A,#01H,OSCILA13
        SETB P2.5                ;SETEO LOS BITS PARA CONFIGURAR EL OSCILADOR
        CLR  P2.6                ;EN XT (CRISTAL STANDAR)
        LJM  FINOSCILA1
OSCILA13:
        CLR  P2.5                ;SETEO LOS BITS PARA CONFIGURAR EL OSCILADOR
        CLR  P2.6                ;EN LP (CRISTAL DE BAJA FRECUENCIA)
FINOSCILA1:

```

```

; PONGO LAS SEÑALES NECESARIAS PARA PROGRAMAR LA PALABRA
        JNB  BSERIAL,$           ;ESPERO INTERRUPCION SERIAL (DATO)
        CLR  BSERIAL            ;BORRO LA BANDERA SERIAL
        CLR  P1.1               ;PONGO EL VOLTAJE DE PROGRAMACION
        LCALL RET1US            ;ANCHO DEL PULSO EN ALTO
        NOP
        CLR  P0.2               ;PONGO EN BAJO TOCK1
        NOP
        LCALL RET10MS          ;ANCHO DEL PULSO EN BAJO (PROGRAMANDO)
        NOP
        SETB P0.2               ;PONGO EN ALTO TOCK1
        MOV  A,#OFFH
        MOV  P2,A
        MOV  A,#0EFH
        MOV  P0,A
        LCALL RET10US          ;ANCHO DEL PULSO EN ALTO (VERIFICACION)
        CLR  P0.2               ;PONGO EN BAJO TOCK1
        LCALL RET10US          ;ANCHO DEL PULSO EN BAJO
        LCALL LECDAT18         ;LEO EL DATO
        SETB P0.2               ;PONGO EN ALTO TOCK1
        SETB P1.1               ;DESABILITO VOLTAJE DE PROGRAMACION
        CLR  A
        MOV  P2,A
        MOV  P0,A
        CLR  P1.4
        MOV  A,DSERIAL
        CJNE A,#06H,CONTINU1
        LJM  COMANDOS

```

CONTINU1:

```

;*****
; * SUBROUTINA PARA PROGRAMAR LA MEMORIA EPROM DE LOS PIC16C DE 18 PINES *
;*****
; RECIVO EL NUMERO DE DATOS A GRABAR

```

```

        JNB  BSERIAL,$           ;ESPERO EL NUMERO DE DATOS QUE CONTIENE
        CLR  BSERIAL            ;LA LINEA A SER PROGRAMADA
CONT1:
        MOV  R5,A               ;CARGO EL #DE DATOS QUE CONTIENE LA LINEA
        MOV  SBUF,A             ;RETRANSMITO
        CJNE R5,#00H,SIGP18
        LJM  COMANDOS

```

SIGP18:

```

; ESPERO QUE SE ENVIE DATO DE LA DIRECCION
        LCALL PROGDIR          ;PREPARO LA DIRECCION DONDE EMPIEZA LA
                                ;DIRECCION DEL PROGRAMA A SER PROGRAMADO

```

; RECIVO EL SEPARADOR ENTRE LA DIRECCION Y LOS DATOS

```

JNB   BSERIAL,$
CLR   BSERIAL
MOV   A,DSERIAL
MOV   SBUF,A           ;RETRANSMITO EL SEPARADOR DE DATOS

```

;1.- PONGO EL VOLTAJE DE PROGRAMACION

```

CLR   P1.5
CLR   P1.6
SETB  P1.4           ;CONECTO VDD AL MICRO
SETB  P0.2           ;PONGO EN ALTO TOCK1
MOV   R2,#0FFH
DJNZ  R2,$
CLR   P1.1           ;HABILITO EL VOLTAJE DE PROGRAMACION

```

;2.- SACO LA DIRECCION  
LCALL SACARDIR

;3.- EN ESTE MOMENTO RECIVO LOS DATOS A SER PROGRAMADOS

ESPERA:

```

JNB   BSERIAL,$     ;ESPERO EL BYTE LMS DEL DATO A SER PROGRAMADO
CLR   BSERIAL
LCALL FORMATO

```

;4.- SACO EL DATO A SER PROGRAMADO

```

MOV   A,DSERIAL
RRC   A
MOV   P2.5,C       ;SACO EL DATO AL PORTICO
RRC   A
MOV   P2.6,C
RRC   A
MOV   P0.0,C
RRC   A
MOV   P0.1,C
RRC   A
MOV   P0.5,C
RRC   A
MOV   P0.6,C
RRC   A
MOV   P0.7,C
RRC   A
MOV   P2.0,C
MOV   SBUF,A
JNB   BSERIAL,$     ;ESPERO EL BYTE MSB DEL DATO A SER PROGRAMADO
CLR   BSERIAL
LCALL FORMATO
MOV   A,DSERIAL     ;RECIBO EL BYTE MSB DEL DATO
RRC   A
MOV   P2.1,C       ;SACO AL PORTICO EL DATO
RRC   A
MOV   P2.2,C
RRC   A
MOV   P2.3,C
RRC   A
MOV   P2.4,C
RRC   A
RRC   A
RRC   A
RRC   A

```

```

MOV  SBUF,A
JNB  BSERIAL,$           ;ESPERO RESPUESTA DEL COMPUTADOR
CLR  BSERIAL

```

;5.- PONGO EL PULSO DE PROGRAMACION

```

CLR  P0.2                ;PONGO EN BAJO TOCK1
NOP
LCALL RET10MS
NOP
SETB P0.2
MOV  A,#0FFH
MOV  P2,A
MOV  A,#0EFH
MOV  P0,A
LCALL RET10US

```

;6.- VERIFICACION

```

CLR  P0.2
NOP

```

;7.- LECTURA DEL DATO

```

LCALL LECDAT18          ;LEO EL DATO
NOP
SETB P0.2
NOP
SETB P1.5
NOP
CLR  P1.5
NOP
DJNZ R5,ESPERA         ;ESPERO EL SIGUIENTE DATO A SER PROGRAMADO

```

;8.- RECIBO EL CHEKSUM

```

SETB P1.1
CLR  A
MOV  P2,A
MOV  P0,A
JNB  BSERIAL,$
CLR  BSERIAL
LCALL FORMATO
MOV  A,DSERIAL
MOV  SBUF,A            ;RETRANSMITO EL CHECKSUM

```

;9.- DESABILITO EL VOLTAJE DE PROGRAMACION

```

JNB  BSERIAL,$
CLR  BSERIAL
MOV  A,DSERIAL
CJNE A,#00H,SALTO1     ;SI EL NUMERO DE DATOS A PROGRAMAR ES
                       ;CERO TERMINA LA PROGRAMACION.

CLR  P1.4
LJMP COMANDOS          ;ESPERO COMANDO DE PROGRAMACION/LECTURA
SALTO1:
LJMP CONT1

```

PROGPIC28:

```

;*****
; *          : SUBROUTINA PARA PROGRAMAR LOS MICROS PIC16C DE 28 PINES          *
;*****

```

```

*****
; *          PROGRAM0 LA CONFIGURACION DE LA PALABRA EN LOS PICS DE 28 PINES          *
*****
SETB  P1.4          ;CONECTO VDD AL MICRO
CLR   P1.5          ;PONGO EN BAJO OSC1
SETB  P2.4          ;PONGO EN ALTO TOCK1

MOV   A,COMANDO

; DETERMINO SI HABILITO O NO EL CODIGO DE PROTECCION
JNB   ACC.3,CODEP2  ;DECIDO SI PROGRAMO EL CODIGO DE PROTECCION
SETB  P0.3          ;RA3
LJMP  FINCODE2

CODEP2:          ;HABILITO EL CODIGO DE PROTECCION
CLR   P0.3          ;RA3
FINCODE2:

; DETERMINO SI HABILITO EN WATCHDOGTIMER
MOV   A,COMANDO
JB    ACC.4,WDTE2   ;CHEQUEO SI PROGRAMO EL WATCHDOGTIMER
CLR   P0.2          ;DESABILITO WDTE
LJMP  FINWDTE2

WDTE2:
SETB  P0.2          ;HABILITO WDTE
FINWDTE2:

*****
; *          SETEO QUE TIPO DE OSCILADOR VOY A PROGRAMAR          *
*****
MOV   A,COMANDO
ANL   A,#60H
SWAP  A
RR    A
CJNE  A,#03H,OSCILA21
SETB  P0.1          ;SETEO LOS BITS PARA CONFIGURAR EL OSCILADOR
SETB  P0.0          ;EN RC (OSCILADOR RC)
LJMP  FINOSCILA2

OSCILA21:
CJNE  A,#02H,OSCILA22
SETB  P0.1          ;SETEO LOS BITS PARA CONFIGURAR EL OSCILADOR
CLR   P0.0          ;EN HS (CRISTAL DE ALTA VELOCIDAD)
LJMP  FINOSCILA2

OSCILA22:
CJNE  A,#01H,OSCILA23
CLR   P0.1          ;SETEO LOS BITS PARA CONFIGURAR EL OSCILADOR
SETB  P0.0          ;EN XT (CRISTAL STANDAR)
LJMP  FINOSCILA2

OSCILA23:
CLR   P0.1          ;SETEO LOS BITS PARA CONFIGURAR EL OSCILADOR
CLR   P0.0          ;EN LP (CRISTAL DE BAJA FRECUENCIA)
FINOSCILA2:

; PONGO LAS SEÑALES NECESARIAS PARA PROGRAMAR LA PALABRA
JNB   BSERIAL,$     ;ESPERO INTERRUPCION SERIAL (DATO)
CLR   BSERIAL       ;BORRO LA BANDERA SERIAL
CLR   P1.0          ;PONGO EL VOLTAJE DE PROGRAMACION
NOP
LCALL RET1US        ;ANCHO DEL PULSO EN ALTO
NOP

```



```

CLR    P2.4                ;PONGO EN BAJO TOCK1
NOP
LCALL  RET10MS            ;ANCHO DEL PULSO EN BAJO
NOP
SETB   P2.4                ;PONGO EN ALTO TOCK1
MOV    A,#0FFH
MOV    P2,A
MOV    P0,A
LCALL  RET10US           ;ANCHO DEL PULSO EN ALTO
CLR    P2.4                ;PONGO EN BAJO TOCK1
NOP
LCALL  RET10US           ;ANCHO DEL PULSO EN ALTO
LCALL  LECDAT28
SETB   P2.4                ;PONGO EN ALTO TOCK1
SETB   P1.0                ;DESABILITO VPP
CLR    A
MOV    P2,A
MOV    P0,A
CLR    P1.4                ;DESCONECTO VDD
MOV    A,DSERIAL
CJNE   A,#06H,CONTINU2
LJMP   COMANDOS

```

CONTINU2:

```

; *****
; *   SUBROUTINA PARA PROGRAMAR LA MEMORIA EPROM DE LOS PIC16C DE 28 PINES   *
; *****

```

; RECIVO EL NUMERO DE DATOS A GRABAR

```

JNB    BSERIAL,$
CLR    BSERIAL

```

CONT2:

```

MOV    R5,A                ;CARGO EL #DE DATOS QUE CONTIENE LA LINEA
MOV    SBUF,R5             ;RETRANSMITO
CJNE   R5,#00H,SIGP28
LJMP   COMANDOS

```

SIGP28:

; ESPERO QUE SE ENVIE DATO DE LA DIRECCION  
LCALL PROGDIR

; RECIVO EL SEPARADOR ENTRE LA DIRECCION Y LOS DATOS

```

JNB    BSERIAL,$
CLR    BSERIAL
MOV    A,DSERIAL
MOV    SBUF,A                ;RETRANSMITO EL SEPARADOR DE DATOS

```

;1.- PONGO EL VOLTAJE DE PROGRAMACION

```

CLR    P1.5
SETB   P1.6
SETB   P2.4
SETB   P1.4                ;CONECTO VDD AL MICRO
MOV    R2,#0FFH
DJNZ   R2,$
CLR    P1.0                ;HABILITO EL VOLTAJE DE PROGRAMACION

```

;2.- SACO LA DIRECCION  
LCALL SACARDIR

;3.- EN ESTE MOMENTO RECIVO LOS DATOS A SER PROGRAMADOS

ESPERA1:

```
JNB  BSERIAL,$
CLR  BSERIAL
LCALL FORMATO
```

;4.- SACO EL DATO A SER PROGRAMADO

```
MOV  A,DSERIAL
MOV  P0,A
MOV  SBUF,A
JNB  BSERIAL,$
CLR  BSERIAL
LCALL FORMATO
MOV  A,DSERIAL
ANL  A,#0FH
MOV  P2,A
MOV  SBUF,A
JNB  BSERIAL,$           ;ESPERO RESPUESTA DEL COMPUTADOR
CLR  BSERIAL
```

;5.- PONGO EL PULSO DE PROGRAMACION

```
CLR  P2.4
NOP
LCALL RET10MS
SETB P2.4
LCALL RET10US
MOV  A,#0FFH
MOV  P0,A
MOV  P2,A
```

;6.- VERIFICACION

```
CLR  P2.4
NOP
LCALL RET10US
```

;7.- LEO EL DATO A TRAVES DEL PORTICO P0 Y P2 Y TRANSMITO

```
LCALL LECDAT28
NOP
SETB P2.4
NOP
SETB P1.5
NOP
CLR  P1.5
DJNZ R5,ESPERA1
```

;8.- RECIBO CHEKSUM

```
SETB P1.0
CLR  A
MOV  P2,A
MOV  P0,A
JNB  BSERIAL,$
CLR  BSERIAL
LCALL FORMATO
MOV  A,DSERIAL
MOV  SBUF,A
```

;9.- DESABILITO EL VOLTAJE DE PROGRAMACION

```
JNB  BSERIAL,$
```

```

CLR    BSERIAL
MOV    A,DSERIAL
CJNE  A,#00H,SALTO2
CLR    P1.4                ;CONECTO VDD AL MICRO
LJMP   COMANDOS
SALTO2:
LJMP   CONT2

```

```

;*****
; *      LECTURA DE LA MEMORIA EPROM DE LOS MICROCONTROLADORES PIC      *
;*****

```

LECTURA:

```

LCALL  SELECCION
MOV    R6,#0FFH           ;RESETEO EL BYTE DE DIRECCION LBS
CLR    A                  ;BORRO EL ACUMULADOR
MOV    R7,A              ;RESETEO EL BYTE DE DIRECCION MBS
JB     MICRO,LECTURA1   ;SELECCIONO QUE MICRO VOY A LBER

```

```

;*****
; *      LECTURA DE LA MEMORIA EPROM DEL MICRO DE 18 PINES      *
;*****

```

```

CLR    P1.6
CLR    P1.5
SETB   P1.4              ;HABILITO VDD
MOV    A,#0EFH
MOV    P0,A
MOV    A,#0FFH
MOV    P2,A

```

;1.- SELECCIONO MODO DE VERIFICACION

```

CLR    P1.1              ;HABILITO EL VOLTAJE DE PROGRAMACION

```

;2.- CUANDO TOCK1 Y OSC1 ESTAN EN ALTO EL CONTENIDO DE LA DIRECCION

; SELECCIONADA ES COLOCADA SOBRE EL PORTICO A Y B

READMEMORY:

```

SETB   P1.5              ;PONGO EN ALTO OSC1

```

;3.- LEO EL DATO A TRAVES DEL PORTICO P0 Y P2 Y TRANSMITO

```

LCALL  LECDAT18
CLR    P1.5              ;PONGO EN BAJO OSC1

```

;4.- INCREMENTO LA DIRECCION

```

INC    R6

```

;5.- COMPARO LA DIRECCION CON EL TANAÑO

```

MOV    A,R6
CJNE  A,#0FFH,READMEMORY
SETB   P1.5
LCALL  LECDAT18
CLR    P1.5              ;PONGO EN BAJO OSC1
MOV    R6,#00H
INC    R7
MOV    A,R7
CJNE  A,TAMAÑO,READMEMORY

```

;6.- REGRESO EL ZOCALO A CONDICIONES NORMALES

```

SETB   P1.1              ;DESABILITO VOLTAJE DE PROGRAMACION
CLR    A

```

```

MOV P2,A
MOV P0,A
CLR P1.4
LJMP COMANDOS

```

```

;*****
;*          LECTURA DE LA MEMORIA EPROM DEL MICRO DE 28 PINES          *
;*****

```

LECTURA1:

```

SETB P1.6
CLR P1.5
SETB P2.4
SETB P1.4          ;HABILITO VDD
MOV A,#0FFH
MOV P2,A
MOV P0,A

```

;1.- SELECCIONO MODO DE VERIFICACION/PROGRAMACION

```

CLR P1.0          ;HABILITO EL VOLTAJE DE PROGRAMACION

```

;2.- CUANDO TOCK1 Y OSC1 ESTAN EN ALTO EL CONTENIDO DE LA DIRECCION

; SELECCIONADA ES COLOCADA SOBRE EL PORTICO A Y B

READMEMORY1:

```

SETB P1.5          ;PONGO EN ALTO A OSC1

```

;3.- LEO EL DATO A TRAVES DEL PORTICO P0 Y P2 Y TRANSMITO

```

LCALL LECDAT28

```

```

CLR P1.5

```

;4.- INCREMENTO LA DIRECCION

```

INC R6

```

;5.- COMPARA LA DIRECCION CON EL TAMAÑO

```

MOV A,R6

```

```

CJNE A,#0FFH,READMEMORY1

```

```

SETB P1.5

```

```

LCALL LECDAT28

```

```

CLR P1.5          ;PONGO EN BAJO OSC1

```

```

MOV R6,#00H

```

```

INC R7

```

```

MOV A,R7

```

```

CJNE A,TAMAÑO,READMEMORY1

```

;6.- REGRESO EL ZOCALO A CONDICIONES NORMALES

```

SETB P1.0          ;DESABILITO VOLTAJE DE PROGRAMACION

```

```

CLR A

```

```

MOV P2,A

```

```

MOV P0,A

```

```

CLR P1.4

```

```

LJMP COMANDOS

```

```

;*****
;*          SUBROUTINA DE LAS CONDICIONES INICIALES DEL SOCKET DE PROGRAMACION          *
;*****

```

CONDICION:

```

MOV A,#0FH

```

```

MOV P1,A

```

```

CLR A

```

```

MOV P2,A
MOV P0,A
RET

```

```

;*****
;* RECIVO LA DIRECCION EN DONDE VA A COMENZAR LA PROGRAMACION *
;*****

```

```

PROGDIR:

```

```

    JNB  BSERIAL,$           ;ESPERO EL BYTE MAS SIGNIFICATIVO DE LA
                           ;DIRECCION DE INICIO DE PROGRAMACION

```

```

    CLR  BSERIAL
    LCALL FORMATO
    MOV  A,DSERIAL
    MOV  R7,A

```

```

                           ;CARGO EN R7 CON BYTE MAS SIGNIFICATIVO DE LA
                           ;DIRECCION DE INICIO DE PROGRAMACION
    MOV  SBUF,A             ;RETRANSMITO

```

```

    JNB  BSERIAL,$           ;ESPERO EL BYTE MENOS SIGNIFICATIVO DE LA
                           ;DIRECCION DE INICIO DE PROGRAMACION

```

```

    CLR  BSERIAL
    LCALL FORMATO
    MOV  A,DSERIAL
    MOV  R6,A

```

```

                           ;CARGO EN R6 CON EL BYTE MENOS SIGNIFICATIVO
                           ;DE LA DIRECCION DE INICIO DE PROGRAMACION
    MOV  SBUF,A             ;RETRANSMITO
    RET

```

```

;*****
;* RUTINA PARA PODER DAR LOS PULSOS NECESARIOS EN OSC1 PARA PODER UBICAR *
;* LA DIRECCION DE INICIO DE PROGRAMACION DE LA EPROM DE LOS MICROS *
;*****

```

```

SACARDIR:

```

```

    LCALL PULSOSC1
    CJNE R7,#00H,DIR1
    CJNE R6,#00H,DIR
    LJMP FINDIR

```

```

DIR:

```

```

    LCALL PULSOSC1
    DJNZ R6,DIR
    LJMP FINDIR

```

```

DIR1:

```

```

    CJNE R6,#00H,PUL
    LCALL PULSOSC1
    DEC  R7
    DEC  R6
    CJNE R7,#00H,PUL
    LJMP DIR

```

```

PUL:

```

```

    LCALL PULSOSC1
    DJNZ R6,PUL
    DEC  R6
    DEC  R7
    CJNE R7,#00H,PUL
    LJMP DIR

```

```

FINDIR:

```

```

    RET

```

```

;*****
;* RUTINA QUE PERMITE DAR UN PULSO EN EL PIN OSC1 PARA INCREMENTAR PC *
;*****

```

```

PULSOSC1:
    SETB  P1.5
    MOV   R3,#04H
    DJNZ  R3,$
    CLR   P1.5
    RET

```

```

;*****
; *           SUBROUTINA PARA LEER EL DATO DEL MICRO DE 18 PINES           *
;*****

```

```

LECDAT18:
    MOV   C,P2.5
    RRC   A
    MOV   C,P2.6
    RRC   A
    MOV   C,P0.0
    RRC   A
    MOV   C,P0.1
    RRC   A
    MOV   C,P0.5
    RRC   A
    MOV   C,P0.6
    RRC   A
    MOV   C,P0.7
    RRC   A
    MOV   C,P2.0
    RRC   A
    MOV   SBUF,A           ;TRANSMITO LSB DEL DATO
    ; ESPERO RESPUESTA DEL COMPUTADOR
    JNB   BSERIAL,$
    CLR   BSERIAL
    MOV   A,P2
    RR    A
    ANL   A,#0FH
    MOV   SBUF,A           ;TRASMITO MSB DEL DATO
    ; ESPERO RESPUESTA DEL COMPUTADOR
    JNB   BSERIAL,$
    CLR   BSERIAL
    RET

```

```

;*****
; *           SUBROUTINA PARA LEER EL DATO DEL MICRO DE 28 PINES           *
;*****

```

```

LECDAT28:
    MOV   A,P2           ;DATO MSB
    MOV   R1,A
    MOV   A,P0           ;DATO LSB
    MOV   SBUF,A         ;TRANSMITO DATO LSB
    ; ESPERO RESPUESTA DEL COMPUTADOR
    JNB   BSERIAL,$
    CLR   BSERIAL
    MOV   A,R1
    ANL   A,#0FH         ;DATO MSB
    MOV   SBUF,A         ;TRANSMITO DATO MSB
    ; ESPERO RESPUESTA DEL COMPUTADOR
    JNB   BSERIAL,$
    CLR   BSERIAL
    RET

```

```

;*****
;*
;* SELECCIONO EL DISPOSITIVO CON EL QUE SE VA A TRABAJAR *
;*****
SELECCION:
    MOV  A,COMANDO           ;MUEVO EL COMANDO AL ACUMULADOR
    JB   ACC.0,SELEC1        ;CHEQUEO CON CUAL DE LOS MICROS SE VA
                                ;A TRABAJAR

; SE HA SELECCIONADO EL MICRO DE 18 PINES CON ACC.0 = 0
    CLR  MICRO               ;BORRO BANDERA MICRO
    SETB P1.0                ;DESABILITO Vpp PARA 28 PINES
    CLR  P1.3
    CLR  P0.4
    SETB P1.2                ;DESABILITO LA LINEA P03 --- RA3
    CLR  P1.6
    CLR  P1.7
    LCALL TAMAÑOS           ;LLAMO A LA RUTINA PARA DETERMINAR EL
                                ;TAMAÑO DE LA MEMORIA

    LMP  FINSELEC

SELEC1:
; SE HA SELECCIONADO EL MICRO DE 28 PINES CON ACC.0 = 1
    SETB MICRO               ;SETEO BANDERA MICRO
    SETB P1.1                ;DESABILITO Vpp PARA 18 PINES
    SETB P1.3                ;DESABILITO GND PARA MICRO DE 18 PINES
    SETB P0.4
    CLR  P1.2                ;CONECTO LA LINEA P0.3 --- RA3
    SETB P1.6
    SETB P1.7
    LCALL TAMAÑOS           ;LLAMO A LA RUTINA PARA DETERMINAR EL
                                ;TAMAÑO DE LA MEMORIA

FINSELEC:
    RET

;*****
;*
;* DETERMINO EL TAMAÑO DE LA MEMORIA DEL MICRO *
;*****
TAMAÑOS:
    MOV  A,COMANDO
    JB   ACC.1,MEMO1K
    JB   ACC.2,MEMO2K
    MOV  TAMAÑO,#02H
    LMP  FINTAMAÑO

MEMO1K:
    MOV  TAMAÑO,#04H
    LMP  FINTAMAÑO

MEMO2K:
    MOV  TAMAÑO,#08H

FINTAMAÑO:
    RET

;*****
;*
;* RETARDO DE UN MICROSEGUNDO *
;*****
RET1US:
    NOP
    RET

```

```

*****
; *                               RETARDO DE 10 MILISEGUNDOS                               *
; *****
RET10MS:
    MOV    R0,#0CH
ESP:
    MOV    B,#0FFH
    DJNZ   B,$
    DJNZ   R0,ESP
    RET

*****
; *                               RETARDO DE 10 MICROSEGUNDOS                               *
; *****
RET10US:
    MOV    R0,#02H
    DJNZ   R0,$
    RET

FORMATO:
    MOV    B,A
    MOV    SBUF,A
    JNB    BSERIAL,$           ;ESPERO INTERRUPCION SERIAL (DATO)
    CLR    BSERIAL           ;BORRO LA BANDERA SERIAL
    MOV    A,DSERIAL
    SWAP   A
    ADD    A,B
    MOV    DSERIAL,A
    RET

*****
; *                               RUTINA A LA INTERRUPCION SERIAL                               *
; *****
SERIAL:
    PUSH   PSW
    JB     TI,FINSERIAL
    MOV    A,SBUF
    MOV    DSERIAL,A
    SETB   BSERIAL
FINSERIAL:
    CLR    TI
    CLR    RI
    POP    PSW
    RETI

END

```



# LISTADO DEL PROGRAMA PARA EL COMPUTADOR EN VISUAL BASIC

## MODULE - 1

'Variable utilizada para presentar el tipo de Microcontrolador

Global PIC As String

Global PIC1 As String

'Variable utilizada para el portico de comunicaciones

Global PORT As String

Global PORT1 As String

'Variable utilizada para el tipo de oscilador que utiliza el microcontrolador

Global OSC As String

Global OSC1 As String

'Variable utilizada para determinar si esta habilitada el Watchdog Timer

Global WDT As String

Global WDT1 As String

'Variable utilizada para determinar si esta habilitado el código de protección

Global CP As String

Global CP1 As String

'Variables para el control de la grilla que presenta el buffer de memoria

Global IROW As Integer

Global iCol As Integer

Global CRLF

'Variable en que se guarda el path así como también el nombre del archivo.hex

Global archivo

'Variable que determina el tamaño de la memoria

Global Tamaño As Integer

'FV Valor del número de filas que contienen error en la verificación

Global FV As Integer

'Variables para poder determinar el valor del comando a

'transmitir, así como también el comando que nos permite determinar

'si existe error en la verificación de la configuración del

'microcontrolador

Global Comando As Integer

Global Comando1 As Integer

Global Control1 As Integer

Global Control2 As Integer

Global Control3 As Integer

Global Control4 As Integer

Global Control5 As Integer

Global Control6 As Integer

' Vector para guardar los datos del archivo condificado

' para tener datos de 12 bits

Public Dato!\$()

' Vector para guardar los datos del archivo a codificar

Public Dato2\$()

' Vector para guardar los datos provenientes de la lectura del micro para su verificaci3/4n  
Public Dato3\$()

' Vector para guardar los datos provenientes de la lectura del micro para su verificaci3/4n  
Public Dato4\$()

' Vector para guardar los datos provenientes de la lectura del micro para su verificaci3/4n  
Public Dato5\$()

Public av()  
Public Config\$(2)  
Public ConfigV\$

'variable utilizada para el control de errores  
'del programa  
Global Errora  
Global Errors  
Global Errorcs  
Global ErrorPic  
Global ErrorConfig

'Variable que se utiliza en la edici3/4n del buffer de memoria  
Global Edit  
Global EDITAR  
Global Verificacion  
Global Band  
Global ID

Sub GuardarArchivo()  
I = 0  
If Edit = 1 Then  
    Edit = 0  
    frmPic.CMDialog1.filename = "C:\progpic.hex"  
Else  
End If  
  
If frmPic.CMDialog1.filename = "" Then  
    GoTo Fin  
Else  
    Open frmPic.CMDialog1.filename For Output As #1  
    frmPic.Grid1.Col = 0  
  
    For fila = 1 To IROW - 1  
        frmPic.Grid1.Row = fila  
        If Val("&H" + frmPic.Grid1.Text) = 0 Then  
            direcc1\$ = frmPic.Grid1.Text  
        Else  
            direcc\$ = Val("&H" + frmPic.Grid1.Text) \* 2  
            direcc1\$ = Hex(direcc\$)  
            l = Len(direcc1\$)  
            If l = 2 Then  
                direcc1\$ = "00" + direcc1\$  
            Else  
                direcc1\$ = "0" + direcc1\$  
            End If  
        End If  
    End If

```

Print #1, ":";
Print #1, "10";
Print #1, direcc1$;
Print #1, "00";

For columna = 1 To 16
    frmPic.Grid1.Col = columna
    If Len(frmPic.Grid1.Text) = 4 Then
        d1$ = Mid(frmPic.Grid1.Text, 3, 2)
        d2$ = Mid(frmPic.Grid1.Text, 1, 2)
    ElseIf Len(frmPic.Grid1.Text) = 3 Then
        d1$ = Mid(frmPic.Grid1.Text, 2, 2)
        d2$ = Mid(frmPic.Grid1.Text, 1, 1)
    End If

    Print #1, d1$;

    If Len(d2$) = 1 Then
        Print #1, "0" + d2$;
    Else
        Print #1, d2$;
    End If

    If columna = 8 Then
        Print #1, "00";
        Print #1, Chr(13) + Chr(10);
        Bandera = 1
    ElseIf columna = 16 Then
        Print #1, "00";
        Print #1, Chr(13) + Chr(10);
    End If

    If Bandera = 1 Then
        Bandera = 0
        Print #1, ":";
        Print #1, "10";
        If Val("&H" + direcc1$) = 0 Then
            Print #1, "0010";
        Else
            direcc2$ = Hex(Val("&H" + direcc1$) + 16)
            l = Len(direcc2$)
            If l = 2 Then
                direcc2$ = "00" + direcc2$
            Else
                direcc2$ = "0" + direcc2$
            End If

            Print #1, direcc2$;
        End If
        Print #1, "00";
    Else
        End If
    End If
Next columna

frmPic.Grid1.Col = 0
Next fila
End If
Print #1, ":";
Print #1, "0000000";

```

```
Print #1, "1FF";
Close #1
Fin:
End Sub
```

#### **frmAbout - 1**

```
Private Sub cmdOK_Click()
Hide      "Descargamos de la memoria el formulario About
End Sub
```

```
Private Sub Form_Load ()
End Sub
```

#### **frmAyuda - 1**

```
Private Sub Form_Load()
Open "c:\progpic\ayuda.txt" For Input As #5
frmAyuda.Text1.Text = Input$(LOF(5), #5)
Close #5
End Sub
```

```
Private Sub mnuSalir_Click()
frmAyuda.Visible = False
End Sub
```

```
Private Sub Text1_KeyUp(KeyCode As Integer, Shift As Integer)
Open "c:\progpic\ayuda.txt " For Input As #5
frmAyuda.Text1.Text = Input$(LOF(5), #5)
Close #5
End Sub
```

#### **frmConfigBits - 1**

```
Private Sub cmdAceptar_Click()
OSC1 = OSC
If OSC1 = "RC" Then
    Control5 = 6
    ElseIf OSC1 = "XT" Then
        Control5 = 2
        ElseIf OSC1 = "LP" Then
            Control5 = 0
            ElseIf OSC1 = "HS" Then
                Control5 = 4
            End If
WDT1 = WDT
If WDT1 = "ON" Then
    Control4 = 1
Else
    Control4 = 0
End If

CP1 = CP
If CP1 = "ON" Then
    Control3 = 0
Else
    Control3 = 8
End If
ErrorPic = 1

Hide
End Sub
```

```

Private Sub cmdCancelar_Click()
    OSC = OSC1
    WDT = WDT1
    CP = CP1
    ErrorPic = 1
    Unload Me
End Sub

Private Sub chkProtecCode_Click(Value As Integer)
    If chkProtecCode.Value = True Then
        CP = "ON"
    Else
        CP = "OFF"
    End If
End Sub

Private Sub chkWatch_Click(Value As Integer)
    If chkWatch.Value = True Then
        WDT = "ON"
    Else
        WDT = "OFF"
    End If
End Sub

Private Sub Form_Load()
    If OSC1 = "RC" Then
        optOsc(0).Value = True
    ElseIf OSC1 = "XT" Then
        optOsc(1).Value = True
    ElseIf OSC1 = "LP" Then
        optOsc(2).Value = True
    ElseIf OSC1 = "HS" Then
        optOsc(3).Value = True
    End If

    If WDT1 = "ON" Then
        chkWatch.Value = True
    Else
        chkWatch.Value = False
    End If

    If CP1 = "ON" Then
        chkProtecCode.Value = True
    Else
        chkProtecCode.Value = False
    End If
End Sub

Private Sub optOsc_Click(INDEX As Integer, Value As Integer)
    Select Case INDEX
        Case 0
            OSC = "RC"
        Case 1
            OSC = "XT"
        Case 2
            OSC = "LP"
        Case 3
            OSC = "HS"
    End Select

```

End Sub

**frmEditarBuffer - 1**

Dim Char

Dim Char1

Dim Key

Dim Key1

Dim Bandera

Dim Bandera1

Private Sub Command1\_Click()

Dato1\$(Val("&H" + Direccion.Text)) = UCase(Dato.Text)

direcc = Val("&H" + Direccion.Text) \* 2

If Len(Dato.Text) = 3 Then

    d1\$ = UCase(Mid\$(Dato.Text, 2, 2))

    d2\$ = "0" + UCase(Mid\$(Dato.Text, 1, 1))

ElseIf Len(Dato.Text) = 2 Then

    d1\$ = UCase(Mid\$(Dato.Text, 1, 2))

    d2\$ = "00"

ElseIf Len(Dato.Text) = 1 Then

    d1\$ = "0" + UCase(Dato.Text)

    d2\$ = "00"

End If

Dato2\$(direcc) = d1\$

Dato2\$(direcc + 1) = d2\$

' LAS SIGUIENTES INSTRUCCIONES NOS PERMITE CARGAR LA

' SIGUIENTE DIRECCION DEL BUFFER CON SU RESPECTIVO DATO

If EDITAR = 0 Then

I = Hex("&H" + Direccion.Text) + 1)

Else

I = 0

EDITAR = 0

End If

Direccion.Text = I

j = Val("&H" + Direccion.Text)

If j = Tamaño Then

EDITAR = 1

Else

End If

Dato.Text = Dato1\$(j)

ErrorPic = 1

Edit = 0

Hide

End Sub

Private Sub Command2\_Click()

ErrorPic = 0

Edit = 1

Unload Me

End Sub

Private Sub Dato\_Change()

If Bandera1 = 1 Then

```

    Bandera1 = 0
    MsgBox "No es un valor Hexadecimal", vbCritical, "ERROR"
    Dato.Text = ""
Else
End If
End Sub

```

```

Private Sub Dato_KeyPress(KeyAscii As Integer)
    Char1 = Chr(KeyAscii)
    Key1 = Asc(UCase(Char1))
    If Key1 > 64 And Key1 < 71 Then
        Bandera1 = 0
        ElseIf Key1 > 47 And Key1 < 58 Then
            Bandera1 = 0
            ElseIf Key1 = 8 Then
                Dato.Text = ""
                ElseIf Key1 = 13 Then
                    If Dato.Text = "" Then
                        Dato.Text = "000"
                    Else
                        Command1.SetFocus
                    End If
                Else
                    Bandera1 = 1
                End If
            End Sub

```

```

Private Sub Direccion_Change()
    If Bandera = 1 Then
        Bandera = 0
        MsgBox "Error en el ingreso del valor", vbCritical, "ERROR"
        Direccion.Text = ""
    Else
    End If
End Sub

```

```

Private Sub Direccion_KeyPress(KeyAscii As Integer)
    Char = Chr(KeyAscii)
    Key = Asc(UCase(Char))
    If Key > 64 And Key < 71 Then
        Bandera = 0
        ElseIf Key > 47 And Key < 58 Then
            Bandera = 0
            ElseIf Key = 8 Then
                Direccion.Text = ""
                ElseIf Key = 13 Then
                    If Direccion.Text = "" Then
                        Direccion.Text = "000"
                    Else
                        Dato.SetFocus
                    End If
                Else
                    Bandera = 1
                End If
            End Sub

```

```

Private Sub Direccion_LostFocus()
    If Val("&H" + Direccion.Text) > Tama±o Then
        MsgBox "El valor de la Direccion excede el tama±o" & Chr$(13) & "de la memoria del Microcontrolador",

```

```

vbCritical, "ERROR"
Direccion.Text = Hex(Tama±o)
Else

End If
End Sub

Private Sub Form_Load()
Dato.Text = Dato1$(0)
End Sub

frmMicro - 1
Private Sub cmdAceptar_Click()
PIC1 = PIC
If PIC1 = "16C54" Then
    Control1 = 0
    Control2 = 0
    IROW = 33
    Tama±o = 511
ElseIf PIC1 = "16C54A" Then
    Control1 = 0
    Control2 = 0
    IROW = 33
    Tama±o = 511
ElseIf PIC1 = "16C55" Then
    Control1 = 1
    Control2 = 0
    IROW = 33
    Tama±o = 511
ElseIf PIC1 = "16C56" Then
    Control1 = 0
    Control2 = 2
    IROW = 65
    Tama±o = 1023
ElseIf PIC1 = "16C57" Then
    Control1 = 1
    Control2 = 4
    IROW = 129
    Tama±o = 2047
ElseIf PIC1 = "16C58A" Then
    Control1 = 0
    Control2 = 4
    IROW = 129
    Tama±o = 2047
End If

ReDim Preserve Dato1$(Tama±o)
For I = 0 To Tama±o
    Dato1$(I) = Hex$(4095)
Next I
ErrorPic = 0
Hide
End Sub

Private Sub cmdCancelar_Click()
PIC = PIC1
ErrorPic = 1
Unload Me
End Sub

```



```

Private Sub Form_Load()
If PIC = "16C54" Then
    optPic(0).Value = True
    ElseIf PIC1 = "16C54A" Then
        optPic(1).Value = True
        ElseIf PIC1 = "16C55" Then
            optPic(2).Value = True
            ElseIf PIC1 = "16C56" Then
                optPic(3).Value = True
                ElseIf PIC1 = "16C57" Then
                    optPic(4).Value = True
                    ElseIf PIC1 = "16C58A" Then
                        optPic(5).Value = True

End If
End Sub

```

```

Private Sub optPic_Click(INDEX As Integer, Value As Integer)
Select Case INDEX
    Case 0
        PIC = "16C54"
    Case 1
        PIC = "16C54A"
    Case 2
        PIC = "16C55"
    Case 3
        PIC = "16C56"
    Case 4
        PIC = "16C57"
    Case 5
        PIC = "16C58A"
End Select
End Sub

```

### **frmPic - 1**

```

Dim Dato
Dim Dato$
Dim DatoRecivido

```

```

Sub Comunicacion()
On Error GoTo errorcomserial
TEST = 6
Rem -----
Rem -   Seteo del Puerto de Comunicaciones Serial -
Rem -----

```

```

Serial.Settings = "9600,n,8,1"
Serial.InputLen = 0
Serial.PortOpen = True
Serial.Output = Chr$(TEST)

```

```

Do
    Dummy = DoEvents()
    I = I + 1
    If I = 3000 Then
        Band = 1
        Exit Do
    Else
        End If
End If

```

```
Loop Until Serial.InBufferCount >= 0
I = 0
```

```
If Band = 1 Then GoTo fin1
  Respuesta = Serial.Input
  respuesta1 = Asc(Respuesta)
  Print respuesta1
  If respuesta1 = TEST Then
    MsgBox "Comunicacion O.K.", vbInformation, "PROGRAMADOR RESPONDE"
    Errors = 0
    mnuPortico.Enabled = False
    cmdPortico.Enabled = False
  Else
    Errors = 1
    MsgBox "Falla de Comunicacion a travÚz del PÓrtico serial" + Chr$(13) + PORT + " Dato recibido no coincide"
  End If
  Serial.PortOpen = False
  Exit Sub
```

```
fin1:
  Band = 0
  Errors = 1
  MsgBox "Con el Programador a travÚz del PÓrtico Serial " + PORT + Chr$(13) + "Revisar si esta conectado el Programador al Computador" + Chr$(13) + "o si el Programador esta prendido", vbInformation, "ERROR DE COMUNICACION ....."
  Serial.PortOpen = False
  Exit Sub
```

```
errorcomserial:
```

```
  MsgBox "El Portico de Comunicaci3n Serial " + "" + PORT + "" + Chr$(13) + "no esta disponible. " + Chr$(13) + Chr$(13) + " Utilice otro PÓrtico Serial", vbCritical, "ERROR DE COMUNICACION"
```

```
  mnuPortico.Enabled = True
  cmdPortico.Enabled = True
  frmPortico.Show 1
  If Errors = 1 Then
    Errors = 1
    Exit Sub
  Else
    Resume
  End If
End Sub
```

```
Sub AbrirArchivo()
```

```
  On Error GoTo fin1
```

```
  CMDialog1.CancelError = True
```

```
  CMDialog1.Action = 1
```

```
  If CMDialog1.filename = "" Then
```

```
    Errora = 1
```

```
    GoTo Fin
```

```
  Else
```

```
    ReDim Preserve Dato2$(2 * Tamaño + 1)
```

```
    For I = 0 To 2 * Tamaño + 1
```

```
      Dato2$(I) = Hex$(255)
```

```
    Next I
```

```
    Abrir$ = CMDialog1.filename
```

```
    Open Abrir$ For Input As #1
```

```

Do
If EOF(1) Then
Exit Do
Else
Y1$ = Input(1, 1) 'leo los dos puntos
Y2$ = Input(2, 1) 'leo el numero de datos
Numdatos = Val("&H" + Y2$)
If Numdatos = 0 Then
Exit Do
Else

Y3$ = Input(4, 1) 'leo la direcci3n de inicio
Direccion = Val("&H" + Y3$)
Y4$ = Input(2, 1) 'leo el separador

For dat = 1 To Numdatos
Y5$ = Input(2, 1) 'leo los datos
DIREC = Direccion + dat - 1
Dato2$(DIREC) = Y5$
Next dat

Y6$ = Input(2, 1) 'leo checksum
Y7$ = Input(2, 1) 'leo salto de lÍnea
End If
End If
Loop
Close #1

ReDim Preserve Dato1$(Tama±o)
For I = 0 To Tama±o
Dato1$(I) = Hex$(Val("&h" + Dato2$(2 * I)) + Val("&H" + Dato2$(2 * I + 1)) * 256)
If Dato1$(I) = Hex$(65535) Then
Dato1$(I) = Hex$(4095)
Else
End If
Next I
Fin:
End If
Exit Sub

fin1:
Errora = 1
Exit Sub
Close #1
End Sub

Sub LeerMemoria()
On Error GoTo FinL
Cls
If Verificacion = 1 Then
Mensaje = "VERIFICAR"
Else
Mensaje = "LEER"
End If

Respuesta = MsgBox("Inserte el CHIP en el Zocalo", vbQuestion + vbOKCancel, "DESEA " + Mensaje + " LA
MEMORIA")
If Respuesta = vbCancel Then GoTo Fin
If Verificacion = 1 Then

```

```

    lblinfo(0).Caption = "Verificando"
Else
    lblinfo(0).Caption = "Leyendo"
End If

frmPic.Enabled = False

Rem -----
Rem - Comando es el dato que se va a enviar al microcontrolador -
Rem -----

    Comando1 = Control1 + Control2 + Control3
    Comando2 = Control4 + Control5

Rem -----
Rem - Preparo el Paquete de Comunicaciones -
Rem -----

Serial.InputLen = 0
Label2.Caption = "PORT : " + PORT
Serial.PortOpen = True

Rem -----
Rem - ENVIO EL COMANDO AL MICROCONTROLADOR -
Rem -----

Serial.Output = Chr$(Comando1)
Do
    Dummy = DoEvents()
    j = j + 1
    If j = 2000 Then GoTo Finel
    Loop Until frmPic.Serial.InBufferCount >= 1
    j = 0
    recep = Serial.Input
    recep = Asc(recep)
    If recep = Comando1 Then GoTo Sigal
    MsgBox "Resetea el Programador", vbInformation
    GoTo Finel
Sigal:
Serial.Output = Chr$(Comando2)
Do
    Dummy = DoEvents()
    j = j + 1
    If j = 2000 Then GoTo Finel
    Loop Until frmPic.Serial.InBufferCount >= 1
    j = 0
    recep = Serial.Input
    recep = Asc(recep)
If recep = Comando1 + Comando2 * 16 Then GoTo Sigal2
    MsgBox "Resetea el Programador", vbInformation
    GoTo Finel
Sigal2:
Serial.Output = Chr$(0)
' RECIVO LOS DATOS DE LA CONFIGURACION DEL MICRO

For C = 1 To 2
Do
    Dummy = DoEvents()
    j = j + 1

```

```

    If j = 2000 Then GoTo Finel
Loop Until frmPic.Serial.InBufferCount >= 1
j = 0
recep = Serial.Input
recep = Asc(recep)
Config$(C) = Hex(recep)
Serial.Output = Chr$(0)
Next C

If Len(Config$(1)) = 2 Then
    Config$(1) = Mid(Config$(1), 2, 1)
Else
End If

For I = 0 To Tama#o
Do
    Dummy = DoEvents()
    j = j + 1
    If j = 2000 Then GoTo Finel
Loop Until frmPic.Serial.InBufferCount >= 1
j = 0
recep = Serial.Input
recep = Asc(recep)
Serial.Output = Chr$(0)
Do
    Dummy = DoEvents()
    j = j + 1
    If j = 2000 Then GoTo Finel
Loop Until frmPic.Serial.InBufferCount >= 1
j = 0
Recep1 = Serial.Input
Recep1 = Asc(Recep1)
Recep1 = Recep1 * 256
Recep2 = recep + Recep1
If Verificacion = 0 Then
lblinfo(1).Caption = Hex(I)
lblinfo(2).Caption = Hex(Recep2)
If Recep2 = 65535 Then
    Recep2 = 4095
Else
End If
Else
If Recep2 = 65535 Then
    Recep2 = 4095
Else
End If
End If

If Verificacion = 0 Then
    Dato1$(I) = Hex(Recep2)
Else
    If Recep2 = 4095 Then
        MemoBlank = 1
    Else
        MemoBlank = 0
    End If
End If
' Indico al programador que el dato fue recibido
Serial.InputLen = 0

```

```

Serial.Output = Chr$(0)
Next I

lblinfo(1).Caption = ""
lblinfo(2).Caption = ""

If Verificacion = 0 Then
Else
If MemoBlank = 0 Then
MsgBox "Memoria no se encuentra en Blanco", vbInformation, "Error"
frmPic.Enabled = True
Serial.PortOpen = False
GoTo Fin
Else
MsgBox "Memoria en Blanco", vbInformation, "Verificaci3¼n O.K."
frmPic.Enabled = True
Serial.PortOpen = False
GoTo Fin
End If
End If
Serial.PortOpen = False
frmPic.Enabled = True
GoTo Fin

Finel:
MsgBox "Error en lectura de memoria", vbInformation, "programador responde"
frmPic.Enabled = True
Serial.PortOpen = False
lblinfo(0).Caption = ""
Exit Sub

Fin:
lblinfo(0).Caption = ""
Exit Sub

FinL:
Label2.Caption = "PORT :" + PORT
MsgBox "El Portico de Comunicaci3¼n Serial " + "" + PORT + "" + Chr$(13) + "no esta disponible. " +
Chr$(13) + Chr$(13) + " Utilice otro P3¼rtico Serial", vbCritical, "ERROR DE COMUNICACION"
frmPortico.Show 1
If Errors = 0 Then
frmPic.Enabled = True
lblinfo(0).Caption = ""
Exit Sub
Else
Resume
End If
End Sub

Sub llenado()
Rem -----
Rem - SETEO LA GRILLA -
Rem -----

Grid1.Cols = 34
Grid1.Rows = IROW

Grid1.ColWidth(0) = TextWidth("0000i")
Grid1.Row = 0
Grid1.Col = 0

```

```

Grid1.Text = ""

For C = 1 To 16
    Grid1.ColWidth(C) = TextWidth("000i")
Next C
Grid1.ColWidth(17) = TextWidth("i")
For C = 18 To 33
    Grid1.ColWidth(C) = TextWidth("Q")
Next C

Grid1.Col = 1
I = 0
For a = 1 To 16
    Grid1.Col = a
    Grid1.Text = " 0" & Hex(I)
    I = I + 1
Next a

Rem -----
Rem -   Seteo los valores de la columna 0   -
Rem -----

j = 0
Grid1.Col = 0
For b = 1 To IROW - 1
    Grid1.Row = b
    If j >= 16 Then
        Grid1.Text = "00" & Hex(j)
    Else
        Grid1.Text = "000" & Hex(j)
    End If

    If j >= 256 Then
        Grid1.Text = "0" & Hex(j)
    End If

    j = j + 16
Next b

I = 0
For fila = 1 To IROW - 1
    Grid1.Row = fila
    For columna = 1 To 16
        Grid1.Col = columna
        If Len(Dato1$(I)) = 4 Then
            Grid1.Text = Dato1$(I)
        ElseIf Len(Dato1$(I)) = 3 Then
            Grid1.Text = Dato1$(I)
        ElseIf Len(Dato1$(I)) = 2 Then
            Grid1.Text = "0" + Dato1$(I)
        ElseIf Len(Dato1$(I)) = 1 Then
            Grid1.Text = "00" + Dato1$(I)
        End If

        Grid1.Col = columna + 17
        If Len(Dato1$(I)) >= 3 Then
            If Dato1$(I) = "FFF" Then
                a = 0
            End If
        End If
    Next columna
Next fila

```

```

Else
a = Val("&H" + Mid(Dato1$(I), 2, 2))
End If
Else
a = Val("&H" + Dato1$(I))
End If

If a > 32 Then
b = Chr$(a)
Else
b = "."
End If
Grid1.Text = b
I = I + 1
Next columna
Next fila
Grid1.Visible = True
End Sub

Private Sub cmdAbrir_Click()
AbrirArchivo
If Errora = 1 Then
Errora = 0
Else
llenado
End If
End Sub

Private Sub cmdAbrir_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
labelayuda.Caption = "Permite Abrir un Archivo HEX"
End Sub

Private Sub cmdDispositivo_Click()
frmConfigBits.Show 1
End Sub

Private Sub cmdDispositivo_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
labelayuda.Caption = "Selecciona Bits de Configuraci3n"
End Sub

Private Sub cmdConfiguracion_Click()
frmConfigBits.Show 1
End Sub

Private Sub cmdConfiguracion_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
labelayuda.Caption = "Permite seleccionar los bits de configuraci3n"
End Sub

Private Sub cmdGuardar_Click()
On Error GoTo fmg
CMDialog1.CancelError = True
CMDialog1.Action = 2
GuardarArchivo
Exit Sub
fmg:
Exit Sub

End Sub

```



```

Private Sub cmdGuardar_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
labelayuda.Caption = "Permite Guardar el contenido del buffer de memoria en un Archivo"
End Sub

Private Sub cmdMicro_Click()
frmMicro.Show 1
End Sub

Private Sub cmdMicro_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
labelayuda.Caption = "Permite seleccionar el PIC a programar "
End Sub

Private Sub cmdPortico_Click()
frmPortico.Show 1
End Sub

Private Sub cmdPortico_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
labelayuda.Caption = "Selecciona el P³artico de Comunicaciones"
End Sub

Private Sub cmdProgramar_Click()
frmMicro.Show 1
End Sub

Private Sub cmdProgramar_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
labelayuda.Caption = "Selecciono Microcontrolador"
End Sub

Private Sub Command1_Click()
End Sub

Private Sub Form_Activate()
Rem -----
Rem - PRESENTA LA CONFIGURACION DEL SISTEMA -
Rem -----
Label1.Caption = "PIC   :" + PIC
Label2.Caption = "PORT  :" + PORT
Label3.Caption = "OSC   :" + OSC
Label4.Caption = "WDT   :" + WDT
Label5.Caption = "CP    :" + CP

ReDim Dato3$(Tama±o)
ReDim Dato4$(Tama±o)
ReDim Dato5$(Tama±o)
ReDim av(Tama±o)

'Comando1 es utilizado para verificar la configuracion del
'microcontrolador

Comando = Control3 + Control4 + Control5
If Comando = 0 Then
    ConfigV$ = "0"
    ElseIf Comando = 8 Then
        ConfigV$ = "8"
    ElseIf Comando = 1 Then
        ConfigV$ = "4"
    ElseIf Comando = 9 Then
        ConfigV$ = "C"
    ElseIf Comando = 2 Then

```

```

ConfigV$ = "2"
ElseIf Comando = 10 Then
    ConfigV$ = "A"
ElseIf Comando = 3 Then
    ConfigV$ = "6"
ElseIf Comando = 11 Then
    ConfigV$ = "E"
ElseIf Comando = 4 Then
    ConfigV$ = "1"
ElseIf Comando = 12 Then
    ConfigV$ = "9"
ElseIf Comando = 5 Then
    ConfigV$ = "5"
ElseIf Comando = 13 Then
    ConfigV$ = "D"
ElseIf Comando = 6 Then
    ConfigV$ = "3"
ElseIf Comando = 14 Then
    ConfigV$ = "B"
ElseIf Comando = 7 Then
    ConfigV$ = "7"
ElseIf Comando = 15 Then
    ConfigV$ = "F"

End If

If ErrorPic = 0 Then
    ErrorPic = 1
llenado
GuardarArchivo
Else
End If
End Sub

Private Sub Form_Load()
Rem -----
Rem - Seteo como debe aparecer inicialmente el formulario principal -
Rem -----
Grid1.Visible = False
CRLF = Chr$(13) + Chr$(10)

ReDim Dato1$(Tama±o)
For I = 0 To Tama±o
    Dato1$(I) = Hex$(4095)
Next I

ReDim Dato2$(2 * Tama±o + 1)
For I = 0 To 2 * Tama±o + 1
    Dato2$(I) = Hex$(255)
Next I

Rem -----
Rem Seteamos la configuraci³n para el cuadro de dialogo -
Rem -----

CMDialog1.Filter = "Archivos HEX (*.Hex)|*.Hex|"

frmPic.Visible = True
Comunicacion
Rem llenado

```

```

End Sub

Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
labelayuda.Caption = ""
End Sub

Private Sub Frame3D1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
labelayuda.Caption = "Presenta la configuraci3n escojida para la Programaci3n"
End Sub

Private Sub Grid1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
labelayuda.Caption = "Muestra el Contenido de la Memoria del Microcontrolador"
End Sub

Private Sub imgAbrir_Click()
End Sub

Private Sub imgGuardar_Click()
End Sub

Private Sub imgSerial_Click()
End Sub.

Private Sub Label1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
labelayuda.Caption = "Presenta la configuraci3n escojida para la Programaci3n"
End Sub

Private Sub Label2_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
labelayuda.Caption = "Presenta la configuraci3n escojida para la Programaci3n"
End Sub

Private Sub Label3_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
labelayuda.Caption = "Presenta la configuraci3n escojida para la Programaci3n"
End Sub

Private Sub Label4_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
labelayuda.Caption = "Presenta la configuraci3n escojida para la Programaci3n"
End Sub

Private Sub Label5_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
labelayuda.Caption = "Presenta la configuraci3n escojida para la Programaci3n"
End Sub

Private Sub mnuAbrir_Click()
AbrirArchivo
If Errora = 1 Then
    Errora = 0
Else
    llenado
End If
End Sub

Private Sub mnuAcerca_Click()
frmAbout.Show vbModal
End Sub

Private Sub mnuAyuda1_Click()
frmAyuda.Show 1
End Sub

```

```

Private Sub mnuBits_Click()
frmConfigBits.optOsc(0) = True
frmConfigBits.chkWatch.Value = True
frmConfigBits.chkProtecCode.Value = False
Label1.Caption = "PIC  :" + PIC
Label2.Caption = "PORT :" + PORT
Label3.Caption = "OSC  :" + OSC
Label4.Caption = "WDT  :" + WDT
Label5.Caption = "CP   :" + CP
Control5 = 6
Control4 = 1
Control3 = 8
ConfigV$ = "F"
Comando = Control5 + Control4 + Control3
End Sub

```

```

Private Sub mnuBorrarB_Click()
CMDialog1.filename = ""
Rem -----
Rem - Lleno la Grilla con valores FFF -
Rem -----

```

```

ReDim Dato1$(Tama±o)
For I = 0 To Tama±o
    Dato1$(I) = Hex$(4095)
Next I
llenado
End Sub

```

```

Private Sub mnuComuni_Click()
'If Errorcs = 0 Then
' MsgBox "Ya se a logrado establecer comunicaci³n" + Chr$(13) + "a travÚs del P³rtico Serial " + PORT,
vbInformation, "COMUNICACION"
'Else
    Comunicacion
'End If
End Sub

```

```

Private Sub mnuConfig_Click()
frmConfigBits.Show 1
End Sub

```

```

Private Sub mnuEditarB_Click()
Inicio:
frmEditarBuffer.Show 1
If Edit = 0 Then GoTo Inicio
End Sub

```

```

Private Sub mnuGuardar_Click()
GuardarArchivo
End Sub

```

```

Private Sub mnuGuardarComo_Click()
CMDialog1.Action = 2
If CMDialog1.filename = "" Then
    GoTo Fin
Else
    GuardarArchivo
Fin:

```

```
End If
End Sub
```

```
Private Sub mnuLeer_Click()
```

```
If Errors = 1 Then
```

```
MsgBox "Que no se ha logrado establecer comunicaci3n" + Chr$(13) + "con el Programador", vbInformation, "NO SE PUEDE LEER LA MEMORIA DEBIDO A..."
```

```
Exit Sub
```

```
Else
```

```
Verificacion = 0
```

```
LeerMemoria
```

```
Edit = 1
```

```
If Config$(1) = "0" Then
```

```
frmConfigBits.optOsc(2) = True
```

```
frmConfigBits.chkWatch.Value = False
```

```
frmConfigBits.chkProtecCode.Value = True
```

```
Control5 = 0
```

```
Control4 = 0
```

```
Control3 = 0
```

```
ElseIf Config$(1) = "1" Then
```

```
frmConfigBits.optOsc(1) = True
```

```
frmConfigBits.chkWatch.Value = False
```

```
frmConfigBits.chkProtecCode.Value = True
```

```
Control5 = 2
```

```
Control4 = 0
```

```
Control3 = 0
```

```
ElseIf Config$(1) = "2" Then
```

```
frmConfigBits.optOsc(3) = True
```

```
frmConfigBits.chkWatch.Value = False
```

```
frmConfigBits.chkProtecCode.Value = True
```

```
Control5 = 4
```

```
Control4 = 0
```

```
Control3 = 0
```

```
ElseIf Config$(1) = "3" Then
```

```
frmConfigBits.optOsc(0) = True
```

```
frmConfigBits.chkWatch.Value = False
```

```
frmConfigBits.chkProtecCode.Value = True
```

```
Control5 = 6
```

```
Control4 = 0
```

```
Control3 = 0
```

```
ElseIf Config$(1) = "4" Then
```

```
frmConfigBits.optOsc(2) = True
```

```
frmConfigBits.chkWatch.Value = True
```

```
frmConfigBits.chkProtecCode.Value = True
```

```
Control5 = 0
```

```
Control4 = 1
```

```
Control3 = 0
```

```
ElseIf Config$(1) = "5" Then
```

```
frmConfigBits.optOsc(1) = True
```

```
frmConfigBits.chkWatch.Value = True
```

```
frmConfigBits.chkProtecCode.Value = True
```

```
Control5 = 2
```

```
Control4 = 1
```

```
Control3 = 0
```

```
ElseIf Config$(1) = "6" Then
```

```
frmConfigBits.optOsc(3) = True
```

```
frmConfigBits.chkWatch.Value = True
```

```
frmConfigBits.chkProtecCode.Value = True
```

```

Control5 = 4
Control4 = 1
Control3 = 0
ElseIf Config$(1) = "7" Then
    frmConfigBits.optOsc(0) = True
    frmConfigBits.chkWatch.Value = True
    frmConfigBits.chkProtecCode.Value = True
    Control5 = 6
    Control4 = 1
    Control3 = 0
ElseIf Config$(1) = "8" Then
    frmConfigBits.optOsc(2) = True
    frmConfigBits.chkWatch.Value = False
    frmConfigBits.chkProtecCode.Value = False
    Control5 = 0
    Control4 = 0
    Control3 = 8
ElseIf Config$(1) = "9" Then
    frmConfigBits.optOsc(1) = True
    frmConfigBits.chkWatch.Value = False
    frmConfigBits.chkProtecCode.Value = False
    Control5 = 2
    Control4 = 0
    Control3 = 8
ElseIf Config$(1) = "A" Then
    frmConfigBits.optOsc(3) = True
    frmConfigBits.chkWatch.Value = False
    frmConfigBits.chkProtecCode.Value = False
    Control5 = 4
    Control4 = 0
    Control3 = 8
ElseIf Config$(1) = "B" Then
    frmConfigBits.optOsc(0) = True
    frmConfigBits.chkWatch.Value = False
    frmConfigBits.chkProtecCode.Value = False
    Control5 = 6
    Control4 = 0
    Control3 = 8
ElseIf Config$(1) = "C" Then
    frmConfigBits.optOsc(2) = True
    frmConfigBits.chkWatch.Value = True
    frmConfigBits.chkProtecCode.Value = False
    Control5 = 0
    Control4 = 1
    Control3 = 8
ElseIf Config$(1) = "D" Then
    frmConfigBits.optOsc(1) = True
    frmConfigBits.chkWatch.Value = True
    frmConfigBits.chkProtecCode.Value = False
    Control5 = 2
    Control4 = 1
    Control3 = 8
ElseIf Config$(1) = "E" Then
    frmConfigBits.optOsc(3) = True
    frmConfigBits.chkWatch.Value = True
    frmConfigBits.chkProtecCode.Value = False
    Control5 = 4
    Control4 = 1
    Control3 = 8

```

```

ElseIf Config$(1) = "F" Then
    frmConfigBits.optOsc(0) = True
    frmConfigBits.chkWatch.Value = True
    frmConfigBits.chkProtecCode.Value = False
    Control5 = 6
    Control4 = 1
    Control3 = 8
End If

Comando = Control5 + Control4 + Control3
llenado

Label1.Caption = "PIC  :" + PIC
Label2.Caption = "PORT :" + PORT
Label3.Caption = "OSC  :" + OSC
Label4.Caption = "WDT  :" + WDT
Label5.Caption = "CP   :" + CP

Rem GuardarArchivo
End If
End Sub

Private Sub mnuMicro_Click()
frmMicro.Show 1
End Sub

Private Sub mnuPortico_Click()
frmPortico.Show 1
End Sub

Private Sub mnuProgramar_Click()
' Abro el archivo a ser programado
If CMDialog1.filename = "" Then
    MsgBox "Que no se ha seleccionado el archivo.hex para ser programado", vbInformation, "NO SE PUEDE
EMPESAR A PROGRAMAR DEBIDO A..."
    Exit Sub
Else
End If

On Error GoTo Finp
Respuesta = MsgBox("Inserte el CHIP en el Zocalo", vbQuestion + vbOKCancel, "DESEA PROGRAMAR")

If Respuesta = vbCancel Then
    GoTo Fin
Else
Rem -----
Rem - Preparo el P3/4rtico de Comunicaciones -
Rem -----

Serial.InputLen = 0
Serial.PortOpen = True

lblinfo(0).Caption = "Programando"
Label2.Caption = "PORT :" + PORT
frmPic.Enabled = False

Rem -----
Rem Comando es el dato que se va a enviar al microcontrolador -
Rem -----

```

```

Control6 = 8
Comando1 = Control1 + Control2 + Control3
Comando2 = Control4 + Control5 + Control6
Cls
Print Comando1; Comando2
Rem -----
Rem -   ENVIO EL COMANDO AL MICROCONTROLADOR   -
Rem -----
Serial.Output = Chr$(Comando1)
Do
    Dummy = DoEvents()
    I = I + 1
    If I = 2000 Then GoTo Finep
Loop Until frmPic.Serial.InBufferCount >= 1
I = 0
recep = Serial.Input
recep = Asc(recep)
If recep = Comando1 Then GoTo siga
MsgBox "Reseteo el Programador", vbInformation
GoTo Finep
siga:
Serial.Output = Chr$(Comando2)
Do
    Dummy = DoEvents()
    I = I + 1
    If I = 2000 Then GoTo Finep
Loop Until frmPic.Serial.InBufferCount >= 1
I = 0
recep = Serial.Input
recep = Asc(recep)
Cls
Print recep
If recep = Comando1 + Comando2 * 16 Then GoTo Sigal
MsgBox "Reseteo el Programador", vbInformation
GoTo Finep
Sigal:
Serial.Output = Chr$(0)
Serial.Output = Chr$(0)
'RECIVO LA CONFIGURACION DEL MICROCONTROLADOR
For C = 1 To 2
Do
    Dummy = DoEvents()
    I = I + 1
    If I = 2000 Then GoTo Finep
Loop Until frmPic.Serial.InBufferCount >= 1
I = 0
'Leo el contenido del buffer de recepci3/4n
recep = Serial.Input
recep = Asc(recep)
Config$(C) = Hex(recep)
Serial.Output = Chr$(0)
Next C
Cls
Print Config$(1)
If Len(Config$(1)) = 2 Then
    Config$(1) = Mid(Config$(1), 2, 1)
Else
End If

```



```

'Validación de configuración del microcontrolador
Print ConfigV$
If ConfigV$ = Config$(1) Then
Else
GoTo Finep
End If

d = 0
ReDim dat(2)
Open CMDialog1.filename For Input As #2
Inicio:
  Bandera = 1
  If EOF(2) Then GoTo finOKp
  ' Leo los dos puntos
  Y1$ = Input(1, 2)
  Prog = 0
Inicio1:
  Prog = Prog + 1

  Select Case Prog
  Case 1
  ' Leo los dos datos consecutivos que me dan el numero de datos
  ' que contiene la linea
  Y2$ = Input(2, 2)
  Y = Val("&H" + Y2$)
  Y = Y / 2

  Case 2
  ' Leo los cuatro datos consecutivos que me dan la direccion
  ' de inicio de programacion
  Y2$ = Input(4, 2)
  Y = Val("&H" + Y2$)
  Y = Y / 2
  Y = Hex(Y)
  If Len(Y) = 3 Then
    Y = "0" + Y
  ElseIf Len(Y) = 2 Then
    Y = "00" + Y
  ElseIf Len(Y) = 1 Then
    Y = "000" + Y
  End If
  Y3 = Val("&H" + Mid(Y, 1, 1))
  Y4 = Val("&H" + Mid(Y, 2, 1))
  Y5 = Val("&H" + Mid(Y, 3, 1))
  Y6 = Val("&H" + Mid(Y, 4, 1))

  Case 3
  ' Leo los dos datos consecutivos, son la separacion
  ' entre la direccion y los datos a ser programados
  Y2$ = Input(2, 2)
  Y = Val("&H" + Y2$)

  Case Is > 3
  ' Leo los dos datos consecutivos, son la separacion
  ' entre la direccion y los datos a ser programados
  Y2$ = Input(2, 2)
  U = Val("&H" + Mid(Y2$, 2, 1))
  v = Val("&H" + Mid(Y2$, 1, 1))
  End Select

```

```

If Bandera = 1 Then
  If Y = 0 Then
    Serial.Output = Chr$(Y)
    If banderae = 1 Then GoTo Finep
    GoTo finOKp
  Else
    Bandera = 0
  End If
Else
End If

If Y2$ = Chr$(13) + Chr$(10) Then
  d = 0
  GoTo Inicio
Else
  ' Transmito el dato

  Select Case Prog

  Case 1
    Serial.Output = Chr$(Y)
    ' Espero el dato enviado por el microcontrolador
    Do
      Dummy = DoEvents()
      I = I + 1
      If I = 2000 Then GoTo Finep
    Loop Until frmPic.Serial.InBufferCount >= 1
    I = 0
    recep = Serial.Input
    recep = Asc(recep)
    Cls
    Print recep
    If Y = recep Then GoTo Inicio1
    banderae = 1
    GoTo Inicio1

  Case 2
    Serial.Output = Chr$(Y3)
    ' Espero el dato enviado por el microcontrolador
    Do
      Dummy = DoEvents()
      I = I + 1
      If I = 2000 Then GoTo Finep
    Loop Until frmPic.Serial.InBufferCount >= 1
    I = 0
    recep = Serial.Input
    recep = Asc(recep)

    Serial.Output = Chr$(Y4)
    Do
      Dummy = DoEvents()
      I = I + 1
      If I = 2000 Then GoTo Finep
    Loop Until frmPic.Serial.InBufferCount >= 1
    I = 0
    recep = Serial.Input
    recep = Asc(recep)

```

```

Serial.Output = Chr$(Y5)
Do
  Dummy = DoEvents()
  I = I + 1
  If I = 2000 Then GoTo Finep
Loop Until frmPic.Serial.InBufferCount >= 1
I = 0
recep = Serial.Input
recep = Asc(recep)

Serial.Output = Chr$(Y6)
Do
  Dummy = DoEvents()
  I = I + 1
  If I = 2000 Then GoTo Finep
Loop Until frmPic.Serial.InBufferCount >= 1
I = 0
recep = Serial.Input
recep = Asc(recep)
GoTo Inicio1

```

```

Case 3
Serial.Output = Chr$(Y)
' Espero el dato enviado por el microcontrolador
Do
  Dummy = DoEvents()
  I = I + 1
  If I = 2000 Then GoTo Finep
Loop Until frmPic.Serial.InBufferCount >= 1
I = 0
recep = Serial.Input
recep = Asc(recep)
If Y = recep Then GoTo Inicio1
banderae = 1
GoTo Inicio1

```

```

Case Is > 3
dat(d) = Y
d = d + 1
Serial.Output = Chr$(U)
Do
  Dummy = DoEvents()
  j = j + 1
  If j = 2000 Then GoTo Finep
Loop Until frmPic.Serial.InBufferCount >= 1
j = 0
recep = Serial.Input
recep = Asc(recep)
Serial.Output = Chr$(v)
Do
  Dummy = DoEvents()
  j = j + 1
  If j = 2000 Then GoTo Finep
Loop Until frmPic.Serial.InBufferCount >= 1
j = 0
recep = Serial.Input
recep = Asc(recep)
If d <> 2 Then GoTo Inicio1
Serial.Output = Chr$(0)

```

```

    Dato = dat(0) + dat(1) * 256
    d = 0
    Do
        Dummy = DoEvents()
        j = j + 1
        If j = 2000 Then GoTo Finep
    Loop Until frmPic.Serial.InBufferCount >= 1
    j = 0
    recep = Serial.Input
    recep = Asc(recep)
    Serial.Output = Chr$(0)
    Do
        Dummy = DoEvents()
        j = j + 1
        If j = 2000 Then GoTo Finep
    Loop Until frmPic.Serial.InBufferCount >= 1
    j = 0
    Recep1 = Serial.Input
    Recep1 = Asc(Recep1)
    Serial.Output = Chr$(0)
    Recep1 = Recep1 * 256
    Recep2 = recep + Recep1
    If Dato = Recep2 Then
        GoTo Inicio1
    Else
        banderae = 1
        GoTo Inicio1
    End If
End If

End Select
End If
GoTo Finep
End If

Fin:
    I = 0
    Close #2
    frmPic.Enabled = True
    lblinfo(0).Caption = ""
    Exit Sub

Finep:
    Close #2
    banderae = 0
    I = 0
    frmPic.Enabled = True
    Serial.PortOpen = False
    lblinfo(0).Caption = ""

    MsgBox "Error de Programaci3n", vbCritical
    Exit Sub

finOKp:
    I = 0
    Close #2
    frmPic.Enabled = True
    lblinfo(0).Caption = ""
    Serial.PortOpen = False
    MsgBox "Programacion O.K.", vbInformation, "PROGRAMADOR RESPONDE"

```

```

Exit Sub

Finp:
  Close #2
  Label2.Caption = "PORT :" + PORT
  MsgBox "El Portico de Comunicaci3/n Serial  " + " " + PORT + " " + Chr$(13) + "no esta disponible. " +
Chr$(13) + Chr$(13) + "      Utilice otro P3/artico Serial", vbCritical, "ERROR DE COMUNICACION"
  frmPortico.Show 1
  If Errors = 0 Then
    frmPic.Enabled = True
    lblinfo(0).Caption = ""
    Exit Sub
  Else
    Resume
  End If
End Sub

Private Sub mnuSalir_Click()
If Edit = 1 Then GoTo salir
Kill "c:\progpic.hex"
salir:
End
End Sub

Private Sub mnuVerificar_Click()
Cls
If Errors = 1 Then
  MsgBox "Que no se ha logrado establecer comunicaci3/n" + Chr$(13) + "con el Programador", vbInformation,
"NO SE PUEDE VERIFICAR EL MICRO DEBIDO A..."
  Exit Sub
Else
  On Error GoTo FinV
  FV = 0

Rem -----
Rem - Preparo el P3/artico de Comunicaciones -
Rem -----

  Serial.InputLen = 0
  Serial.PortOpen = True
  Label2.Caption = "PORT :" + PORT

  Respuesta = MsgBox("Inserte el CHIP en el Zocalo", vbQuestion + vbOKCancel, "DESEA VERIFICAR LA
MEMORIA")
  If Respuesta = vbCancel Then GoTo Fin
  lblinfo(0).Caption = "Verificando"
  frmPic.Enabled = False

Rem -----
Rem - Comando es el dato que se va a enviar al microcontrolador -
Rem -----

  Comando1 = Control1 + Control2 + Control3
  Comando2 = Control4 + Control5

Rem -----
Rem - ENVIO EL COMANDO AL MICROCONTROLADOR -
Rem -----

```

```

Serial.Output = Chr$(Comando1)
Do
    Dummy = DoEvents()
    j = j + 1
    If j = 2000 Then GoTo FineV
    Loop Until frmPic.Serial.InBufferCount >= 1
    j = 0
    recep = Serial.Input
    recep = Asc(recep)
    If recep = Comando1 Then GoTo Sigav
    MsgBox "Reseteo el Programador,vbinformation"
    GoTo FineV
Sigav:
Serial.Output = Chr$(Comando2)
Do
    Dummy = DoEvents()
    j = j + 1
    If j = 2000 Then GoTo FineV
    Loop Until frmPic.Serial.InBufferCount >= 1
    j = 0
    recep = Serial.Input
    recep = Asc(recep)
If recep = Comando1 + Comando2 * 16 Then GoTo Sigav1
MsgBox "Reseteo el Programador,vbinformation"
GoTo FineV
Sigav1:
Serial.Output = Chr$(0)
' RECIVO LOS DATOS DE LA CONFIGURACION DEL MICRO
For C = 1 To 2
    Do
        Dummy = DoEvents()
        j = j + 1
        If j = 2000 Then GoTo FineV
        Loop Until frmPic.Serial.InBufferCount >= 1
        j = 0
        recep = Serial.Input
        recep = Asc(recep)
        Config$(C) = Hex(recep)
        Serial.Output = Chr$(0)
    Next C

    If Len(Config$(1)) = 2 Then
        Config$(1) = Mid(Config$(1), 2, 1)
    Else
        End If

    If ConfigV$ = Config$(1) Then
        ErrorConfig = 0
    Else
        ErrorV = 1
        ErrorConfig = 1
    End If

    ReDim av(Tama±o)
    For I = 0 To Tama±o

        Do
            Dummy = DoEvents()
            j = j + 1

```

```

    If j = 2000 Then GoTo FineV
    Loop Until frmPic.Serial.InBufferCount >= 1
    j = 0

    recep = Serial.Input
    recep = Asc(recep)
    Serial.Output = Chr$(0)

    Do
        Dummy = DoEvents()
        j = j + 1
        If j = 2000 Then GoTo FineV
    Loop Until frmPic.Serial.InBufferCount >= 1
    j = 0

    Recep1 = Serial.Input
    Recep1 = Asc(Recep1)
    Recep1 = Recep1 * 256
    Recep2 = recep + Recep1
    Recep2 = Hex(Recep2)

    Dato3$(I) = Recep2
    If Dato3$(I) = Hex$(65535) Then
        Dato3$(I) = Hex$(4095)
    Else
    End If
    If Dato1$(I) = Dato3$(I) Then
    Else
        ErrorV = 1
        av(FV) = Hex(I)
        Dato4$(FV) = Dato1$(I)
        Dato5$(FV) = Dato3$(I)
        FV = FV + 1
    End If
    Serial.Output = Chr$(0)
Next I

If ErrorV = 0 Then
    MsgBox "Verificaci3n O.K.", vbInformation
Else
    MsgBox "Error en la verificaci3n", vbInformation
    frmVerifi.Show 1
End If

End If
GoTo Fin

FineV:
    MsgBox "Error en la verificacion de la memoria", vbInformation, "Error de comunicacion programador responde"
    frmPic.Enabled = True
    Serial.PortOpen = False
    lblinfo(0).Caption = ""
    Exit Sub

Fin:
    frmPic.Enabled = True
    Serial.PortOpen = False
    lblinfo(0).Caption = ""

```

Exit Sub

FinV:

```
Label2.Caption = "PORT :" + PORT
MsgBox "El Portico de Comunicaci3/n Serial " + "" + PORT + "" + Chr$(13) + "no esta disponible. " +
Chr$(13) + Chr$(13) + " Utilice otro P3/artico Serial", vbCritical, "ERROR DE COMUNICACION"
frmPortico.Show 1
If Errors = 0 Then
    frmPic.Enabled = True
    lblinfo(0).Caption = ""
    Exit Sub
Else
    Resume
End If
End Sub
```

Private Sub mnuVerificarB\_Click()

```
If Errores = 1 Then
    MsgBox "Que no se ha logrado establecer comunicaci3/n" + Chr$(13) + "con el Programador", vbInformation,
"NO SE PUEDE VERIFICAR EL MICRO DEBIDO A..."
    Exit Sub
Else
    Verificacion = 1
    LeerMemoria
End If
End Sub
```

Private Sub txtID\_Change()

End Sub

Private Sub txtID\_KeyPress(KeyAscii As Integer)

End Sub

### **frmPortico - 1**

Private Sub cmdAceptar\_Click()

PORT1 = PORT

If PORT1 = "COMM 1" Then

frmPic.Serial.CommPort = 1

ElseIf PORT1 = "COMM 2" Then

frmPic.Serial.CommPort = 2

ElseIf PORT1 = "COMM 3" Then

frmPic.Serial.CommPort = 3

ElseIf PORT1 = "COMM 4" Then

frmPic.Serial.CommPort = 4

End If

frmPic.Label2.Caption = "PORT :" + PORT

ErrorPic = 1

Errors = 1

Errores = 0

Hide

End Sub

Private Sub cmdCancelar\_Click()

PORT = PORT1

ErrorPic = 1

Errors = 0

Errores = 1

Unload Me

End Sub



```

Private Sub optbaud_Click(INDEX As Integer, Value As Integer)
Select Case INDEX
  Case 0
    frmPic.Serial.Settings = 300
  Case 1
    frmPic.Serial.Settings = 600
  Case 2
    frmPic.Serial.Settings = 1200
  Case 3
    frmPic.Serial.Settings = 2400
  Case 4
    frmPic.Serial.Settings = 4800
  Case 5
    frmPic.Serial.Settings = 9600
End Select
End Sub

```

```

Private Sub Form_Load()
If PORT = "COMM 1" Then
  optCom(0).Value = True
ElseIf PORT = "COMM 2" Then
  optCom(1).Value = True
ElseIf PORT = "COMM 3" Then
  optCom(2).Value = True
ElseIf PORT = "COMM 4" Then
  optCom(3).Value = True
End If
End Sub

```

```

Private Sub optCom_Click(INDEX As Integer, Value As Integer)
Select Case INDEX
  Case 0
    PORT = "COMM 1"
  Case 1
    PORT = "COMM 2"
  Case 2
    PORT = "COMM 3"
  Case 3
    PORT = "COMM 4"
End Select
End Sub

```

### **frmPresentacion - 1**

```

Private Sub Command1_Click()
Unload Me
frmPic.Show
End Sub

```

```

Private Sub Form_Load()
frmPresentacion.Move (Screen.Width - Width) / 2, (Screen.Width - Width) / 2
PIC = "16C54"
PORT = "COMM 1"
OSC = "RC"
WDT = "ON"
CP = "OFF"

```

```

Rem -----
Rem - Inicializaci3n de variables auxiliares -
Rem -----

```

```

PIC1 = PIC
PORT1 = PORT
OSC1 = OSC
WDT1 = WDT
CP1 = CP

Rem -----
Rem - Inicializaci3n de Grilla -
Rem -----
IROW = 33
iCol = 16
Tama±o = 511

Rem -----
Rem - Inicializaci3n de Variables de Comando -
Rem -----
Control1 = 0
Control2 = 0
Control3 = 8
Control4 = 1
Control5 = 6
End Sub

frmVerifi - 1
Private Sub Command1_Click()
ErrorV = 0
ErrorConfig = 0
Unload Me
End Sub

Private Sub Form_Load()
GridVerifi.Rows = FV + 1
GridVerifi.Row = 0
GridVerifi.Col = 0
GridVerifi.ColWidth(0) = TextWidth("DIRECCION")
GridVerifi.Text = "Direcci3n"
GridVerifi.Col = 1
GridVerifi.ColWidth(1) = TextWidth("CORRECTO")
GridVerifi.Text = "Correcto"
GridVerifi.Col = 2
GridVerifi.ColWidth(2) = TextWidth("ERRONEO")
GridVerifi.Text = "Erroneo"
For Y = 1 To FV
GridVerifi.Row = Y
GridVerifi.Col = 0
If Len(av(Y - 1)) = 4 Then
GridVerifi.Text = av(Y - 1)
ElseIf Len(av(Y - 1)) = 3 Then
GridVerifi.Text = "0" + av(Y - 1)
ElseIf Len(av(Y - 1)) = 2 Then
GridVerifi.Text = "00" + av(Y - 1)
ElseIf Len(av(Y - 1)) = 1 Then
GridVerifi.Text = "000" + av(Y - 1)
End If
GridVerifi.Col = 1
If Len(Dato4$(Y - 1)) = 4 Then
GridVerifi.Text = Dato4$(Y - 1)
ElseIf Len(Dato4$(Y - 1)) = 3 Then
GridVerifi.Text = Dato4$(Y - 1)

```

```

        ElseIf Len(Dato4$(Y - 1)) = 2 Then
            GridVerifi.Text = "0" + Dato4$(Y - 1)
            ElseIf Len(Dato4$(Y - 1)) = 1 Then
                GridVerifi.Text = "00" + Dato4$(Y - 1)
        End If
        GridVerifi.Col = 2
        If Len(Dato5$(Y - 1)) = 4 Then
            GridVerifi.Text = Dato5$(Y - 1)
            ElseIf Len(Dato5$(Y - 1)) = 3 Then
                GridVerifi.Text = Dato5$(Y - 1)
            ElseIf Len(Dato5$(Y - 1)) = 2 Then
                GridVerifi.Text = "0" + Dato5$(Y - 1)
            ElseIf Len(Dato5$(Y - 1)) = 1 Then
                GridVerifi.Text = "00" + Dato5$(Y - 1)
        End If
    Next Y

If ErrorConfig = 0 Then
    Label1.Visible = False
    Label2.Visible = False
    txtConfig.Visible = False
    txtConfig1.Visible = False
Else
    Label1.Visible = True
    Label2.Visible = True
    txtConfig.Visible = True
    txtConfig1.Visible = True
    txtConfig.Text = "OSC: " + OSC + CRLF + "WDT: " + WDT + CRLF + "CP: " + CP

    If Config$(1) = "0" Then
        txtConfig1.Text = "OSC: LP" + CRLF + "WDT: OFF" + CRLF + "CP: ON"
    ElseIf Config$(1) = "1" Then
        txtConfig1.Text = "OSC: XT" + CRLF + "WDT: OFF" + CRLF + "CP: ON"
    ElseIf Config$(1) = "2" Then
        txtConfig1.Text = "OSC: HS" + CRLF + "WDT: OFF" + CRLF + "CP: ON"
    ElseIf Config$(1) = "3" Then
        txtConfig1.Text = "OSC: RC" + CRLF + "WDT: OFF" + CRLF + "CP: ON"
    ElseIf Config$(1) = "4" Then
        txtConfig1.Text = "OSC: LP" + CRLF + "WDT: ON" + CRLF + "CP: ON"
    ElseIf Config$(1) = "5" Then
        txtConfig1.Text = "OSC: XT" + CRLF + "WDT: ON" + CRLF + "CP: ON"
    ElseIf Config$(1) = "6" Then
        txtConfig1.Text = "OSC: HS" + CRLF + "WDT: ON" + CRLF + "CP: ON"
    ElseIf Config$(1) = "7" Then
        txtConfig1.Text = "OSC: RC" + CRLF + "WDT: ON" + CRLF + "CP: ON"
    ElseIf Config$(1) = "8" Then
        txtConfig1.Text = "OSC: LP" + CRLF + "WDT: OFF" + CRLF + "CP: OFF"
    ElseIf Config$(1) = "9" Then
        txtConfig1.Text = "OSC: XT" + CRLF + "WDT: OFF" + CRLF + "CP: OFF"
    ElseIf Config$(1) = "A" Then
        txtConfig1.Text = "OSC: HS" + CRLF + "WDT: OFF" + CRLF + "CP: OFF"
    ElseIf Config$(1) = "B" Then
        txtConfig1.Text = "OSC: RC" + CRLF + "WDT: OFF" + CRLF + "CP: OFF"
    ElseIf Config$(1) = "C" Then
        txtConfig1.Text = "OSC: LP" + CRLF + "WDT: ON" + CRLF + "CP: OFF"
    ElseIf Config$(1) = "D" Then
        txtConfig1.Text = "OSC: XT" + CRLF + "WDT: ON" + CRLF + "CP: OFF"
    ElseIf Config$(1) = "E" Then
        txtConfig1.Text = "OSC: HS" + CRLF + "WDT: ON" + CRLF + "CP: OFF"

```

```
ElseIf Config$(1) = "F" Then
    txtConfig1.Text = "OSC: RC" + CRLF + "WDT: ON" + CRLF + "CP: OFF"
End If
End If
End Sub
```