Capítulo 4

4. SOFTWARE DE INTEGRACIÓN DE MÓDULOS LÓGICOS.

4. SOFTWARE DE INTEGRACIÓN DE MÓDULOS LÓGICOS.

4.1 Descripción general.

Mediante un programa denominado GEN-ENS_PLM08 desarrollado en Visual Basic ®, se genera de manera automática el código en lenguaje ensamblador que hace posible la traducción de un código en lenguaje SIIL1 a su correspondiente código equivalente en lenguaje ensamblador. Para ello se parte del esqueleto genérico en ensamblador presentado en el capítulo anterior. Este programa en ensamblador generado contendrá componentes fijos y tramos de código que van a corresponder con cada uno de los módulos lógicos que haya declarado el usuario en el programa original. Para generar estos tramos de código se parte de los esqueletos en código ensamblador descritos en el capítulo anterior teniendo cada uno de ellos asociada una matriz que contiene los strings componentes de dicho esqueleto. Algunos de estos componentes son las cadenas mudas cuyo contenido depende de lo declarado por el usuario para cada módulo. En el capítulo anterior se explico para casos específicos la asignación de contenido de esas cadenas mudas.

4.2 Esqueleto de programación y matriz de cadenas asociadas con los módulos lógicos realizables por él PLM08.

El programa GEN_ENS_PLM08 parte de las matrices de cadenas asociadas con cada módulo para generar el código en ensamblador asociado con cada uno de los módulos declarados por el usuario. Para esto llena los espacios correspondientes a las cadenas mudas de acuerdo a los argumentos que contengan los módulos declarados por el usuario, esto para cada uno de los módulos. Para generar el código en ensamblador correspondiente el programa GEN_ENS_PLM08 parte de una función que inserta líneas de lenguaje ensamblador con las cadenas requeridas con el archivo fuente en ensamblador a generar. En cada caso se hace a partir de la matriz de cadenas modificada de acuerdo a lo declarado por el usuario.

En las siguientes secciones se muestra el código esqueleto asociado con cada uno de los módulos y la matriz de cadenas asociada para cada caso. En estas se destacan en negritas las cadenas mudas asociadas.

4.2.1 Esqueleto de programación y matriz de cadenas asociadas con el módulo lógico seguidor.

En la sección 3.2.1 del capítulo anterior se mostró el programa en lenguaje ensamblador asociado con el módulo lógico que realiza un seguidor lógico. El programa se transcribe a continuación:

```
                brclr bitent,grupobitent,seg#00nm
                brset bitent,grupobitent,seg#10nm
seg#00nm:        bclr bitsal,grupobitsal
                bra seg#20nm
seg#10nm:        bset bitsal,grupobitsal
seg#20nm:        nop
```

El esqueleto de programación puede ser construido con un arreglo matricial, donde los distintos elementos (i, j) de la matriz corresponden en posición y orden a los caracteres propios del programa genérico que resuelve cada módulo lógico realizable.

	1	2	3	4	5	6	7	8
1		brclr		**bitent**	,	**grupobitent**	,	seg#00**nm**
2		brset		**bitent**	,	**grupobitent**	,	seg#10**nm**
3	seg#00**nm:**	bclr		**bitsal**	,	**grupobitsal**		
4		bra		seg#20**nm**				
5	seg#10**nm:**	bset		**bitsal**	,	**grupobitsal**		
6	seg#20**nm:**	nop						

Tabla 4.1 Matriz de cadenas asociada con el módulo seguidor lógico.

Después el esqueleto de programación del módulo lógicos realizable por él PLM08 será llamado por el programa GEN_ENS_PLM08 para hacer una sustitución de valores acorde entre los datos ingresados por el usuario al programar en lenguaje SIIL1 y las cadenas mudas vistas en la sección 3.2.1 del capítulo anterior. Generando con ello el programa en lenguaje ensamblador buscado para este caso.

Se hace una sustitución en la matriz de cadenas asociadas, denominada para el caso del código fuente del programa GEN_ENS_PLM08 como StrMatrizEsqueleto. Y una cadena denominada StrModulo. La cadena asociada StrModulo se genera al validar un archivo SIIL1 con el programa GEN_ENS_PLM08; esto para cada módulo lógico declarado por el usuario, (ver sección 4.3 de este capítulo, guía rápida del usuario).

La cadena StrModulo contiene información básica de las cadenas mudas a sustituir, para generar el archivo en código ensamblador para cada módulo lógico.

Por ejemplo; para la siguiente sentencia en SIIL1 que representa la declaración de un módulo seguidor lógico cuya entrada es el bit 7 del grupo 0 de entradas y cuya salida es el bit 0 del grupo de salidas. La sentencia SIIL1 correspondiente es la siguiente:

SEG#1 E07,S00;

Para el caso de este módulo lógico, la cadena StrModulo es:

0101E007S000

Para la cual el primer par de dígitos representa un código numérico, que indica se trata de un módulo seguidor lógico; el siguiente par de dígitos representa el número que el usuario asigno a este módulo; los siguientes tres caracteres indican que la entrada del módulo pertenece al grupo cero de entradas, el siguiente carácter indica que el bit asociado con la entrada es el bit 7. Los siguientes tres caracteres indican que la salida pertenece al grupo cero de salidas y el último carácter indica que el bit físico de salidas, es el bit cero del grupo antes mencionado.

A continuación se muestra en la figura 4.2.2; el código en Visual Basic® que hace la asignación de cadenas mudas en la matriz correspondiente para el caso de un módulo seguidor lógico.

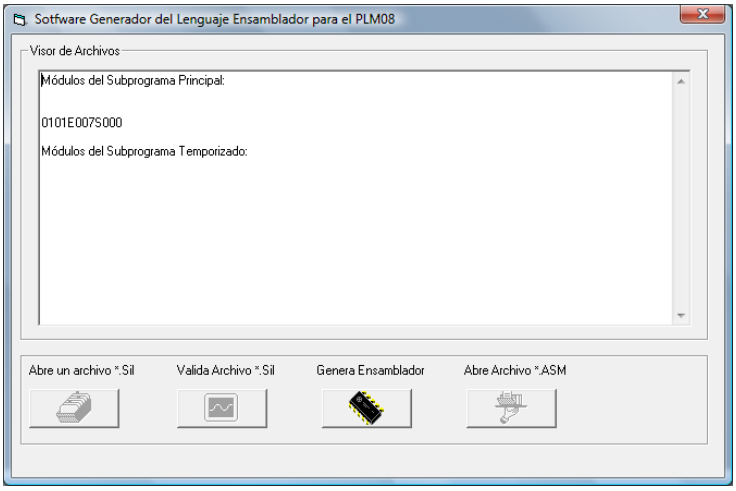


Figura 4.2.1. Ventana que muestra la cadena asociada StrModulo al validar un archivo.

```
Public Sub Seg(StrModulo As String)  'MODULO SEGUIDOR

Dim StrArchivo As String
Dim n As Integer
Dim StrLinea As String
Dim i As Long
Dim j As Long
Dim IntRenglones As Variant
Dim IntColumnas As Variant
Dim StrLineaEns As String

StrArchivo = App.Path & "\Templates\seg.esq"
n = FreeFile()
Open StrArchivo For Input As #n
   Line Input #n, IntRenglones
    For i = 1 To IntRenglones
       Line Input #n, IntColumnas
       For j = 1 To IntColumnas
          Line Input #n, StrLinea
          StrMatrizEsqueleto(i, j) = StrLinea
       Next j
    Next i
Close #n

StrMatrizEsqueleto(1, 4) = Mid(StrModulo, 8, 1)
StrMatrizEsqueleto(1, 6) = "gpe" & LCase(Mid(StrModulo, 6, 2))
StrMatrizEsqueleto(1, 8) = "seg#00" & Mid(StrModulo, 3, 2)
StrMatrizEsqueleto(2, 4) = Mid(StrModulo, 8, 1)
StrMatrizEsqueleto(2, 6) = "gpe" & LCase(Mid(StrModulo, 6, 2))
StrMatrizEsqueleto(2, 8) = "seg#10" & Mid(StrModulo, 3, 2)

StrMatrizEsqueleto(3, 1) = "seg#00" & Mid(StrModulo, 3, 2) & ":        "
StrMatrizEsqueleto(3, 4) = Mid(StrModulo, 12, 1)
StrMatrizEsqueleto(3, 6) = "gps" & LCase(Mid(StrModulo, 10, 2))

StrMatrizEsqueleto(4, 4) = "seg#20" & Mid(StrModulo, 3, 2)
StrMatrizEsqueleto(5, 1) = "seg#10" & Mid(StrModulo, 3, 2) & ":        "
StrMatrizEsqueleto(5, 4) = Mid(StrModulo, 12, 1)
StrMatrizEsqueleto(5, 6) = "gps" & Mid(StrModulo, 10, 2) & "  "
StrMatrizEsqueleto(6, 1) = "seg#20" & Mid(StrModulo, 3, 2) & ":        "

   For i = 1 To 6

       For j = 1 To 8
          StrLineaEns = StrLineaEns & StrMatrizEsqueleto(i, j)
       Next j
       StrLineaEns = StrLineaEns & vbCrLf
   Next i

   StrEnsResultadoMatriz = StrLineaEns

End Sub
```

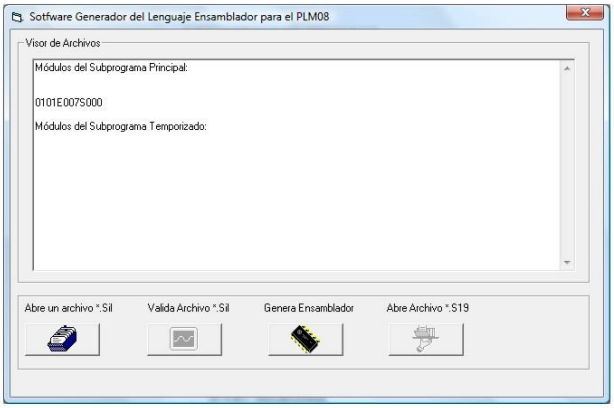


Figura 4.2.2 Código en Visual Basic ® módulo seguidor lógico.

4.2.2 Esqueleto de programación y matriz de cadenas asociadas con el módulo lógico inversor.

En la sección 3.2.2 del capítulo anterior se mostró el programa en lenguaje ensamblador asociado con el módulo lógico que realiza un inversor lógico. El programa se transcribe a continuación:

```
                brclr bitent,grupobitent,inv#00nm
                brset bitent,grupobitent,inv#10nm
inv#00nm:       bset bitsal,grupobitsal
                bra inv#20nm
inv#10nm:       bclr bitsal,grupobitsal
inv#20nm:       nop
```

El esqueleto de programación puede ser construido con un arreglo matricial, donde los distintos elementos (i, j) de la matriz corresponden en posición y orden a los caracteres propios del programa genérico que resuelve cada módulo lógico realizable.

	1	2	3	4	5	6	7	8
1		brclr		**bitent**	,	**grupobitent**	,	inv#00**nm**
2		brset		**bitent**	,	**grupobitent**	,	inv#10**nm**
3	inv#00**nm:**	bset		**bitsal**	,	**grupobitsal**		
4		bra		inv#20**nm**				
5	inv#10**nm:**	bclr		**bitsal**	,	**grupobitsal**		
6	inv#20**nm:**	nop						

Tabla 4.2 Matriz de cadenas asociada con el módulo inversor lógico.

Después el esqueleto de programación del módulo lógico realizable por él PLM08 será llamado por el programa GEN_ENS_PLM08 para hacer una sustitución de valores acorde entre los datos ingresados por el usuario al programar en lenguaje SIIL1 y las cadenas mudas vistas en la sección 3.2.2 del capítulo anterior. Generando con ello el programa en lenguaje ensamblador buscado para este caso.

Se hace una sustitución en la matriz de cadenas asociadas, denominada para el caso del código fuente del programa GEN_ENS_PLM08 como StrMatrizEsqueleto. Y una cadena denominada StrModulo. La cadena asociada StrModulo se genera al validar un archivo SIIL1 con el programa GEN_ENS_PLM08; esto para cada módulo lógico declarado por el usuario, (ver sección 4.3 de este capítulo, guía rápida del usuario).

La cadena StrModulo contiene información básica de las cadenas mudas a sustituir, para generar el archivo en código ensamblador para cada módulo lógico.

Por ejemplo; para la siguiente sentencia en SIIL1 que representa la declaración de un módulo inversor lógico cuya entrada es el bit 7 del grupo 0 de entradas y cuya salida es el bit 0 del grupo de salidas. La sentencia SIIL1 correspondiente es la siguiente:

NOT#1 E07,S00;

Para el caso de este módulo lógico, la cadena StrModulo es:

0201E007S000

Para la cual el primer par de dígitos representa un código numérico, que indica se trata de un módulo inversor lógico; el siguiente par de dígitos representa el número que el usuario asigno a este módulo; los siguientes tres caracteres indican que la entrada del módulo pertenece al grupo cero de entradas, el siguiente carácter indica que el bit asociado con la entrada es el bit 7. Los siguientes tres caracteres indican que la salida pertenece al grupo cero de salidas y el último carácter indica que el bit físico de salidas, es el bit cero del grupo antes mencionado.

A continuación se muestra en la figura 4.2.4; el código en Visual Basic® que hace la asignación de cadenas mudas en la matriz correspondiente para el caso de un módulo inversor lógico.

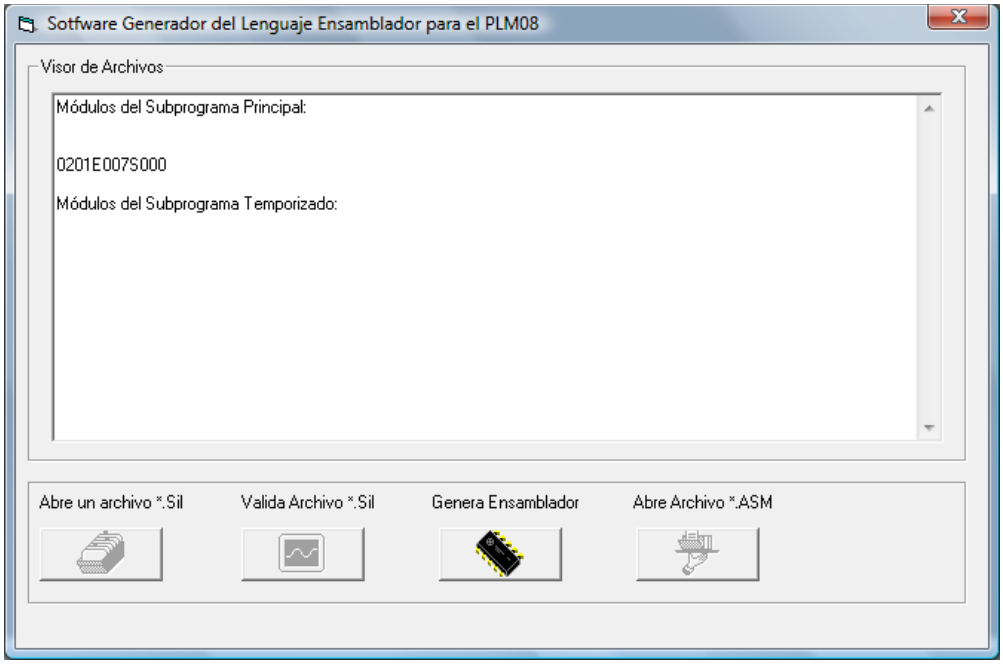


Figura 4.2.3 Ventana que muestra la cadena asociada StrModulo al validar un archivo.

```
Public Sub Inv(StrModulo As String) 'MODULO INVERSOR

Dim StrArchivo As String
Dim n As Integer
Dim StrLinea As String
Dim i As Long
Dim j As Long
Dim IntRenglones As Variant
Dim IntColumnas As Variant
Dim StrLineaEns As String

StrArchivo = App.Path & "\Templates\inv.esq"
n = FreeFile()
Open StrArchivo For Input As #n
   Line Input #n, IntRenglones
    For i = 1 To IntRenglones
       Line Input #n, IntColumnas
       For j = 1 To IntColumnas
          Line Input #n, StrLinea
          StrMatrizEsqueleto(i, j) = StrLinea
       Next j
    Next i
Close #n

StrMatrizEsqueleto(1, 4) = Mid(StrModulo, 8, 1)
StrMatrizEsqueleto(1, 6) = "gpe" & LCase(Mid(StrModulo, 6, 2))
StrMatrizEsqueleto(1, 8) = "inv#00" & Mid(StrModulo, 3, 2)
StrMatrizEsqueleto(2, 4) = Mid(StrModulo, 8, 1)
StrMatrizEsqueleto(2, 6) = "gpe" & LCase(Mid(StrModulo, 6, 2))
StrMatrizEsqueleto(2, 8) = "inv#10" & Mid(StrModulo, 3, 2)

StrMatrizEsqueleto(3, 1) = "inv#00" & Mid(StrModulo, 3, 2) & ":       "
StrMatrizEsqueleto(3, 4) = Mid(StrModulo, 12, 1)
StrMatrizEsqueleto(3, 6) = "gps" & LCase(Mid(StrModulo, 10, 2))

StrMatrizEsqueleto(4, 4) = "inv#20" & Mid(StrModulo, 3, 2)
StrMatrizEsqueleto(5, 1) = "inv#10" & Mid(StrModulo, 3, 2) & ":       "
StrMatrizEsqueleto(5, 4) = Mid(StrModulo, 12, 1)
StrMatrizEsqueleto(5, 6) = "gps" & Mid(StrModulo, 10, 2) & "  "
StrMatrizEsqueleto(6, 1) = "inv#20" & Mid(StrModulo, 3, 2) & ":       "

   For i = 1 To 6

       For j = 1 To 8
          StrLineaEns = StrLineaEns & StrMatrizEsqueleto(i, j)
       Next j
       StrLineaEns = StrLineaEns & vbCrLf
   Next i

   StrEnsResultadoMatriz = StrLineaEns

End Sub
```

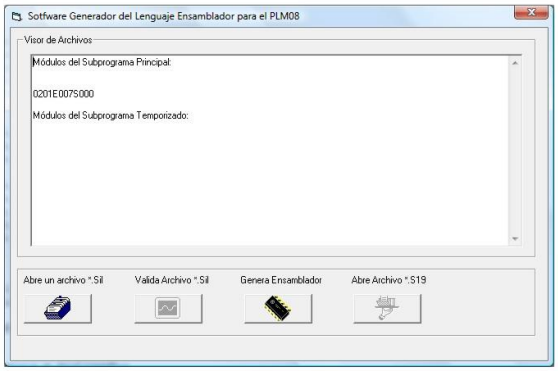


Figura 4.2.4 Código en Visual Basic ® módulo inversor lógico.

4.2.3 Esqueleto de programación y matriz de cadenas asociadas con el módulo lógico de las compuertas lógicas AND, OR, NAND y NOR de dos entradas realizables con el PLM08.

En la sección 3.2.3 del capítulo anterior se mostró el programa en lenguaje ensamblador asociado con el módulo lógico que realiza compuertas lógicas de dos entradas. El programa en el caso de una compuerta AND2 se transcribe a continuación:

```
                clr tesentb
                brclr bitent1,grupobitent1,and2#00nm
                bset 0,tesentb
and2#00nm:      brclr bitent2,grupobitent2,and2#10nm
                bset 1,tesentb
and2#10nm:      lda tesentb
                eor #$XX
                cmp #$YY
                beq and2#20nm
                bclr bitsal,grupobitsal
                bra and2#30nm
and2#20nm:      bset bitsal,grupobitsal
and2#30nm:      nop
```

El esqueleto de programación puede ser construido con un arreglo matricial, donde los distintos elementos (i, j) de la matriz corresponden en posición y orden a los caracteres propios del programa genérico que resuelve cada módulo lógico realizable.

	1	2	3	4	5	6	7	8
1		clr		**tesentb**				
2		brclr		**bitent1**	,	**grupobitent1**	,	and2#00**nm**
3		bset		0	,	tesentb		
4	and2#00**nm**:	brclr		**bitent2**	,	**grupobitent2**	,	and2#10**nm**
5		bset		1	,	tesentb		
6	and2#10**nm**:	lda		tesentb				
7		eor		#$**XX**				
8		cmp		#$**YY**				
9		beq		and2#20**nm**				
10		bclr		**bitsal**	,	**grupobitsal**		
11		bra		and2#30**nm**				
12	and2#20**nm**:	bset		**bitsal**	,	**grupobitsal**		
13	and2#30**nm**:	nop						

Tabla 4.3 Matriz de cadenas asociada con los módulos lógicos AND,NAND, OR, NOR de 2 entradas.

Después el esqueleto de programación del módulo lógico realizable por él PLM08 será llamado por el programa GEN_ENS_PLM08 para hacer una sustitución de valores acorde entre los datos

ingresados por el usuario al programar en lenguaje SIIL1 y las cadenas mudas vistas en la sección 3.2.3 del capítulo anterior. Generando con ello el programa en lenguaje ensamblador buscado para este caso.

Se hace una sustitución en la matriz de cadenas asociadas, denominada para el caso del código fuente del programa GEN_ENS_PLM08 como StrMatrizEsqueleto. Y una cadena denominada StrModulo. La cadena asociada StrModulo se genera al validar un archivo SIIL1 con el programa GEN_ENS_PLM08; esto para cada módulo lógico declarado por el usuario, (ver sección 4.3 de este capítulo, guía rápida del usuario).

La cadena StrModulo contiene información básica de las cadenas mudas a sustituir, para generar el archivo en código ensamblador para cada módulo lógico.

Por ejemplo; para la siguiente sentencia en SIIL1 que representa la declaración de un módulo lógico AND2; cuya primera entrada es el bit 7 del grupo 0 de entradas, su segunda entrada es el bit 6 del grupo 0 de entradas y cuya salida es el bit 1 del grupo de salidas. Con preinversión en ambas entradas de la compuerta.

La sentencia SIIL1 correspondiente es la siguiente:

AND2#1 E07,E06,S01,00;

Para el caso de este módulo lógico, la cadena StrModulo es:

0301E007E006S00100

Para la cual el primer par de dígitos representa un código numérico, que indica se trata de un módulo lógico compuerta AND de dos entradas; el siguiente par de dígitos representa el número que el usuario asigno a este módulo; los siguientes tres caracteres indican que la primera entrada del módulo pertenece al grupo cero de entradas, el siguiente carácter indica que el bit asociado con la primera entrada es el bit 7; los siguientes tres caracteres indican que la segunda entrada del módulo pertenece al grupo cero de entradas, el siguiente carácter indica que el bit asociado con la segunda entrada es el bit 6. Los siguientes tres caracteres indican que la salida pertenece al grupo cero de salidas y el siguiente carácter indica que el bit físico de salidas, es el bit 1 del grupo antes mencionado. Los siguientes y últimos dos dígitos indican que ambas entradas tienen preinversión lógica.A continuación se muestra en la figura 4.2.5; el código en Visual Basic® que hace la asignación de cadenas mudas en la matriz correspondiente para el caso de un módulo lógico de una compuerta AND de dos entradas.

```
Public Sub And2(StrModulo As String)

Dim StrArchivo As String
Dim n As Integer
Dim StrLinea As String
Dim i As Long
Dim j As Long
Dim IntRenglones As Variant
Dim IntColumnas As Variant
Dim StrLineaEns As String

StrArchivo = App.Path & "\Templates\and2.esq"
n = FreeFile()
Open StrArchivo For Input As #n
   Line Input #n, IntRenglones
    For i = 1 To IntRenglones
       Line Input #n, IntColumnas
       For j = 1 To IntColumnas
          Line Input #n, StrLinea
          StrMatrizEsqueleto(i, j) = StrLinea
       Next j
    Next i
Close #n

StrMatrizEsqueleto(2, 4) = Mid(StrModulo, 8, 1)
StrMatrizEsqueleto(2, 6) = "gp" & LCase(Mid(StrModulo, 5, 3))
StrMatrizEsqueleto(2, 8) = "and2#00" & Mid(StrModulo, 3, 2) & "  "
StrMatrizEsqueleto(4, 1) = "and2#00" & Mid(StrModulo, 3, 2) & ":          "
StrMatrizEsqueleto(4, 4) = Mid(StrModulo, 12, 1)

StrMatrizEsqueleto(4, 6) = "gp" & LCase(Mid(StrModulo, 9, 3))
StrMatrizEsqueleto(4, 8) = "and2#10" & Mid(StrModulo, 3, 2) & "  "
StrMatrizEsqueleto(6, 1) = "and2#10" & Mid(StrModulo, 3, 2) & ":          "

If (Mid(StrModulo, 17, 2)) = 0 Then
   StrMatrizEsqueleto(7, 4) = "#$03"
   End If
If (Mid(StrModulo, 17, 2)) = 1 Then
   StrMatrizEsqueleto(7, 4) = "#$02"
   End If
If (Mid(StrModulo, 17, 2)) = 10 Then
   StrMatrizEsqueleto(7, 4) = "#$01"
   End If
If (Mid(StrModulo, 17, 2)) = 11 Then
   StrMatrizEsqueleto(7, 4) = "#$00"
   End If

StrMatrizEsqueleto(9, 4) = "and2#20" & Mid(StrModulo, 3, 2) & "  "
StrMatrizEsqueleto(10, 4) = Mid(StrModulo, 16, 1)
StrMatrizEsqueleto(10, 6) = "gp" & LCase(Mid(StrModulo, 13, 3))
StrMatrizEsqueleto(11, 4) = "and2#30" & Mid(StrModulo, 3, 2) & "  "
StrMatrizEsqueleto(12, 1) = "and2#20" & Mid(StrModulo, 3, 2) & ":          "
StrMatrizEsqueleto(12, 4) = Mid(StrModulo, 16, 1)
StrMatrizEsqueleto(12, 6) = "gp" & LCase(Mid(StrModulo, 13, 3))
StrMatrizEsqueleto(13, 1) = "and2#30" & Mid(StrModulo, 3, 2) & ":          "

   For i = 1 To 13

        For j = 1 To 8
           StrLineaEns = StrLineaEns & StrMatrizEsqueleto(i, j)
        Next j
        StrLineaEns = StrLineaEns & vbCrLf
   Next i

   StrEnsResultadoMatriz = StrLineaEns

End Sub
```

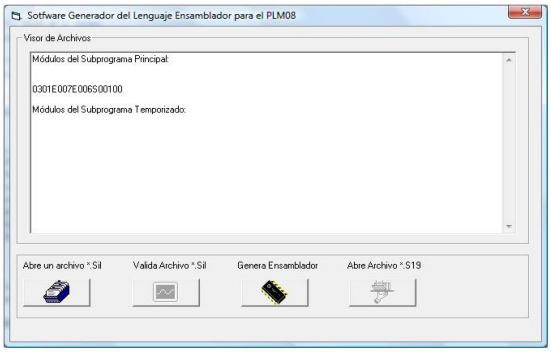


Figura 4.2.5 Código en Visual Basic ® módulo AND 2 entradas y ventana que muestra la cadena asociada StrModulo al validar un archivo

4.2.4 Esqueleto de programación y matriz de cadenas asociadas con el módulo lógico de las compuertas lógicas AND, OR, NAND y NOR de tres entradas realizables con el PLM08.

En la sección 3.2.4 del capítulo anterior se mostró el programa en lenguaje ensamblador asociado con el módulo lógico que realiza compuertas lógicas de tres entradas. El programa en el caso de una compuerta AND3 se transcribe a continuación:

```
                clr tesentb
                brclr bitent1,grupobitent1,and3#00nm
                bset 0,tesentb
and3#00nm:      brclr bitent2,grupobitent2,and3#10nm
                bset 1,tesentb
and3#10nm:      brclr bitent3,grupobitent3,and3#20nm
                bset 2,tesentb
and3#20nm:      lda tesentb
                eor  #$XX
                cmp #$YY
                beq  and3#30nm
                bclr  bitsal,grupobitsal
                bra   and3#40nm
and3#30nm:      bset bitsal,grupobitsal
and3#40nm:      nop
```

El esqueleto de programación puede ser construido con un arreglo matricial, donde los distintos elementos (i, j) de la matriz corresponden en posición y orden a los caracteres propios del programa genérico que resuelve cada módulo lógico realizable.

	1	2	3	4	5	6	7	8
1		clr		**tesentb**				
2		brclr		**bitent1**	,	**grupobitent1**	,	and3#00**nm**
3		bset		0	,	tesentb		
4	and3#00**nm**:	brclr		**bitent2**	,	**grupobitent2**	,	and3#10**nm**
5		bset		1	,	tesentb		
6	and3#10nm:	brclr		**bitent3**	,	**grupobitent3**	,	and3#20**nm**
7		bset		2	,	tesentb		
8	and3#20**nm**:	lda		tesentb				
9		eor		#$**XX**				
10		cmp		#$**YY**				
11		beq		and3#30**nm**				
12		bclr		**bitsal**	,	**grupobitsal**		
13		bra		and3#40**nm**				
14	and3#30**nm**:	bset		**bitsal**	,	**grupobitsal**		
15	and3#40**nm**:	nop						

Tabla 4.4 Matriz de cadenas asociada con los módulos lógicos AND,NAND, OR, NOR de 3 entradas.

Después el esqueleto de programación del módulo lógico realizable por él PLM08 será llamado por el programa GEN_ENS_PLM08 para hacer una sustitución de valores acorde entre los datos ingresados por el usuario al programar en lenguaje SIIL1 y las cadenas mudas vistas en la sección 3.2.4 del capítulo anterior. Generando con ello el programa en lenguaje ensamblador buscado para este caso.

Se hace una sustitución en la matriz de cadenas asociadas, denominada para el caso del código fuente del programa GEN_ENS_PLM08 como StrMatrizEsqueleto. Y una cadena denominada StrModulo. La cadena asociada StrModulo se genera al validar un archivo SIIL1 con el programa GEN_ENS_PLM08; esto para cada módulo lógico declarado por el usuario, (ver sección 4.3 de este capítulo, guía rápida del usuario).

La cadena StrModulo contiene información básica de las cadenas mudas a sustituir, para generar el archivo en código ensamblador para cada módulo lógico.

Por ejemplo; para la siguiente sentencia en SIIL1 que representa la declaración de un módulo lógico AND3; cuya primera entrada es el bit 7 del grupo 0 de entradas, su segunda entrada es el bit 6 del grupo 0 de entradas, su tercera entrada es el bit 5 del grupo 0 de entradas y cuya salida es el bit 1 del grupo de salidas. Con preinversión en sus tres entradas de compuerta.

La sentencia SIIL1 correspondiente es la siguiente:

AND3#1 E07,E06,E05,S01,000;

Para el caso de este módulo lógico, la cadena StrModulo es:

0901E007E006E005S001000

Para la cual el primer par de dígitos representa un código numérico, que indica se trata de un módulo lógico compuerta AND de tres entradas; el siguiente par de dígitos representa el número que el usuario asigno a este módulo; los siguientes tres caracteres indican que la primera entrada del módulo pertenece al grupo cero de entradas, el siguiente carácter indica que el bit asociado con la primera entrada es el bit 7; los siguientes tres caracteres indican que la segunda entrada del módulo pertenece al grupo cero de entradas, el siguiente carácter indica que el bit asociado con la segunda entrada es el bit 6; los siguientes tres caracteres indican que la tercera entrada del módulo pertenece al grupo cero de entradas, el siguiente carácter indica que el bit asociado con la tercera entrada es el bit 5. Los siguientes tres caracteres indican que la salida pertenece al grupo cero de salidas y el siguiente carácter indica que el bit físico de salidas, es el bit 1 del grupo antes mencionado. Los siguientes y últimos tres dígitos indican que todas las entradas tienen preinversión lógica.

A continuación se muestra en la figura 4.2.6; el código en Visual Basic® que hace la asignación de cadenas mudas en la matriz correspondiente para el caso de un módulo lógico de una compuerta AND de tres entradas.

```
Public Sub And3(StrModulo As String)

Dim StrArchivo As String
Dim n As Integer
Dim StrLinea As String
Dim i As Long
Dim j As Long
Dim IntRenglones As Variant
Dim IntColumnas As Variant
Dim StrLineaEns As String


StrArchivo = App.Path & "\Templates\and3.esq"
n = FreeFile()
Open StrArchivo For Input As #n
  Line Input #n, IntRenglones
   For i = 1 To IntRenglones
      Line Input #n, IntColumnas
      For j = 1 To IntColumnas
         Line Input #n, StrLinea
         StrMatrizEsqueleto(i, j) = StrLinea
      Next j
   Next i
Close #n

StrMatrizEsqueleto(2, 4) = Mid(StrModulo, 8, 1)
StrMatrizEsqueleto(2, 6) = "gp" & LCase(Mid(StrModulo, 5, 3))
StrMatrizEsqueleto(2, 8) = "and3#00" & Mid(StrModulo, 3, 2) & " "
StrMatrizEsqueleto(4, 1) = "and3#00" & Mid(StrModulo, 3, 2) & ":          "
StrMatrizEsqueleto(4, 4) = Mid(StrModulo, 12, 1)

StrMatrizEsqueleto(4, 6) = "gp" & LCase(Mid(StrModulo, 9, 3))
StrMatrizEsqueleto(4, 8) = "and3#10" & Mid(StrModulo, 3, 2) & " "
StrMatrizEsqueleto(6, 1) = "and3#10" & Mid(StrModulo, 3, 2) & ":          "
StrMatrizEsqueleto(6, 4) = Mid(StrModulo, 16, 1)

StrMatrizEsqueleto(6, 6) = "gp" & LCase(Mid(StrModulo, 13, 3))
StrMatrizEsqueleto(6, 8) = "and3#20" & Mid(StrModulo, 3, 2) & " "
StrMatrizEsqueleto(8, 1) = "and3#20" & Mid(StrModulo, 3, 2) & ":          "

If (Mid(StrModulo, 21, 3)) = 0 Then
   StrMatrizEsqueleto(9, 4) = "#$07"
   End If
If (Mid(StrModulo, 21, 3)) = 1 Then
   StrMatrizEsqueleto(9, 4) = "#$06"
   End If
If (Mid(StrModulo, 21, 3)) = 10 Then
   StrMatrizEsqueleto(9, 4) = "#$05"
   End If
If (Mid(StrModulo, 21, 3)) = 11 Then
   StrMatrizEsqueleto(9, 4) = "#$04"
End If

If (Mid(StrModulo, 21, 3)) = 100 Then
   StrMatrizEsqueleto(9, 4) = "#$03"
   End If
If (Mid(StrModulo, 21, 3)) = 101 Then
   StrMatrizEsqueleto(9, 4) = "#$02"
   End If
If (Mid(StrModulo, 21, 3)) = 110 Then
   StrMatrizEsqueleto(9, 4) = "#$01"
   End If
If (Mid(StrModulo, 21, 3)) = 111 Then
  StrMatrizEsqueleto(9, 4) = "#$00"
   End If

StrMatrizEsqueleto(11, 4) = "and3#30" & Mid(StrModulo, 3, 2) & " "
StrMatrizEsqueleto(12, 4) = Mid(StrModulo, 20, 1)
StrMatrizEsqueleto(12, 6) = "gp" & LCase(Mid(StrModulo, 17, 3))
StrMatrizEsqueleto(13, 4) = "and3#40" & Mid(StrModulo, 3, 2) & " "
StrMatrizEsqueleto(14, 1) = "and3#30" & Mid(StrModulo, 3, 2) & ":          "
StrMatrizEsqueleto(14, 4) = Mid(StrModulo, 20, 1)
StrMatrizEsqueleto(14, 6) = "gp" & LCase(Mid(StrModulo, 17, 3))
StrMatrizEsqueleto(15, 1) = "and3#40" & Mid(StrModulo, 3, 2) & ":          "

   For i = 1 To 15
      For j = 1 To 8
         StrLineaEns = StrLineaEns & StrMatrizEsqueleto(i, j)
      Next j
      StrLineaEns = StrLineaEns & vbCrLf
   Next i

   StrEnsResultadoMatriz = StrLineaEns

End Sub
```

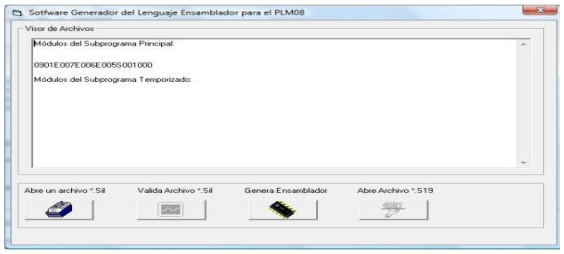


Figura 4.2.6 Código en Visual Basic ® módulo AND 3 entradas y ventana que muestra la cadena asociada StrModulo al validar un archivo.

4.2.5 Esqueleto de programación y matriz de cadenas asociadas con el módulo lógico de las compuertas lógicas AND, OR, NAND y NOR de cuatro entradas realizables con el PLM08.

En la sección 3.2.5 del capítulo anterior se mostró el programa en lenguaje ensamblador asociado con el módulo lógico que realiza compuertas lógicas de cuatro entradas. El programa en el caso de una compuerta AND4 se transcribe a continuación:

```
                clr tesentb
                brclr bitent1,grupobitent1,and4#00nm
                bset 0,tesentb
and4#00nm:        brclr bitent2,grupobitent2,and4#10nm
                bset 1,tesentb
and4#10nm:        brclr bitent3,grupobitent3,and4#20nm
                bset 2,tesentb
and4#20nm:        brclr bitent4,grupobitent4,and4#30nm
                 bset 3,tesentb
and4#30nm:        lda tesentb
                  eor #$XX
                cmp #$YY
                beq and4#40nm
                bclr bitsal,grupobitsal
                bra and4#50nm
and4#40nm:       bset bitsal,grupobitsal
and4#50nm:       nop
```

El esqueleto de programación puede ser construido con un arreglo matricial, donde los distintos elementos (i, j) de la matriz corresponden en posición y orden a los caracteres propios del programa genérico que resuelve cada módulo lógico realizable.

	1	2	3	4	5	6	7	8
1		clr		**tesentb**				
2		brclr		**bitent1**	,	**grupobitent1**	,	and4#00**nm**
3		bset		0	,	tesentb		
4	and4#00**nm**:	brclr		**bitent2**	,	**grupobitent2**	,	and4#10**nm**
5		bset		1	,	tesentb		
6	and4#10nm:	brclr		bitent3	,	grupobitent3	,	and4#20nm
7		bset		2	,	tesentb		
8	and4#20nm	brclr		bitent4	,	grupobitent4	,	and4#30nm
9		bset		3	,	tesentb		
10	and4#30**nm**:	lda		tesentb				
11		eor		#$**XX**				
12		cmp		#$**YY**				

13		beq		and4#40**nm**				
	1	2	3	4	5	6	7	8
14		bclr		**bitsal**	,	**grupobitsal**		
15		bra		and4#50**nm**				
16	and4#40**nm**:	bset		**bitsal**	,	**grupobitsal**		
17	and4#50**nm**:	nop						

Tabla 4.5 Matriz de cadenas asociada con los módulos lógicos AND,NAND, OR, NOR de 4 entradas.

Después el esqueleto de programación del módulo lógico realizable por él PLM08 será llamado por el programa GEN_ENS_PLM08 para hacer una sustitución de valores acorde entre los datos ingresados por el usuario al programar en lenguaje SIIL1 y las cadenas mudas vistas en la sección 3.2.5 del capítulo anterior. Generando con ello el programa en lenguaje ensamblador buscado para este caso.

Se hace una sustitución en la matriz de cadenas asociadas, denominada para el caso del código fuente del programa GEN_ENS_PLM08 como StrMatrizEsqueleto. Y una cadena denominada StrModulo. La cadena asociada StrModulo se genera al validar un archivo SIIL1 con el programa GEN_ENS_PLM08; esto para cada módulo lógico declarado por el usuario, (ver sección 4.3 de este capítulo, guía rápida del usuario).

La cadena StrModulo contiene información básica de las cadenas mudas a sustituir, para generar el archivo en código ensamblador para cada módulo lógico.

Por ejemplo; para la siguiente sentencia en SIIL1 que representa la declaración de un módulo lógico AND4; cuya primera entrada es el bit 7 del grupo 0 de entradas, su segunda entrada es el bit 6 del grupo 0 de entradas, su tercera entrada es el bit 5 del grupo 0, su cuarta entrada es el bit 4 del grupo 0 de entradas y cuya salida es el bit 1 del grupo de salidas. Con preinversión en sus cuatro entradas de compuerta.

La sentencia SIIL1 correspondiente es la siguiente:

AND4#1 E07,E06,E05,E04,S01,0000;

Para el caso de este módulo lógico, la cadena StrModulo es:

1501E007E006E005E004S0010000

Para la cual el primer par de dígitos representa un código numérico, que indica se trata de un módulo lógico compuerta AND de cuatro entradas; el siguiente par de dígitos representa el número que el usuario asigno a este módulo; los siguientes tres caracteres indican que la primera entrada del módulo pertenece al grupo cero de entradas, el siguiente carácter indica que el bit asociado con la primera entrada es el bit 7; los siguientes tres caracteres indican que la segunda entrada del módulo pertenece al grupo cero de entradas, el siguiente carácter indica que el bit asociado con la segunda entrada es el bit 6; los siguientes tres caracteres indican que la tercera

entrada del módulo pertenece al grupo cero de entradas, el siguiente carácter indica que el bit asociado con la tercera entrada es el bit 5; los siguientes tres caracteres indican que la cuarta entrada del módulo pertenece al grupo cero de entradas, el siguiente carácter indica que el bit asociado con la cuarta entrada es el bit 4. Los siguientes tres caracteres indican que la salida pertenece al grupo cero de salidas y el siguiente carácter indica que el bit físico de salidas, es el bit 1 del grupo antes mencionado. Los siguientes y últimos cuatro dígitos indican que todas las entradas tienen preinversión lógica.

A continuación se muestra en la figura 4.2.8; el código en Visual Basic® que hace la asignación de cadenas mudas en la matriz correspondiente para el caso de un módulo lógico de una compuerta AND de cuatro entradas.

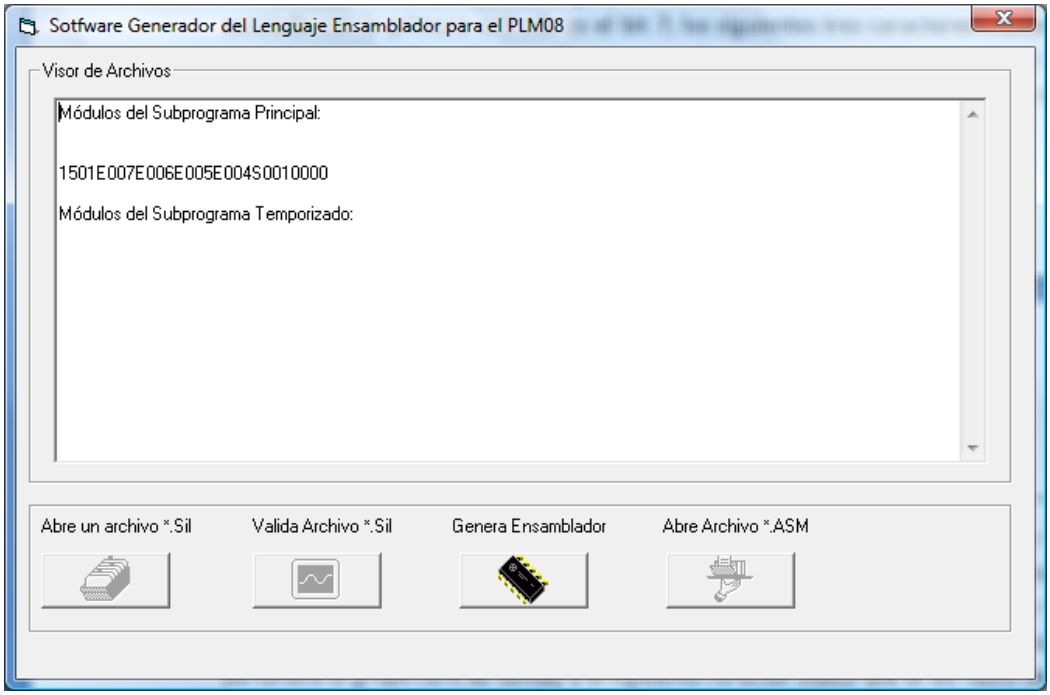


Figura 4.2.7. Ventana que muestra la cadena asociada StrModulo al validar un archivo.

```
Public Sub And4(StrModulo As String)

Dim StrArchivo As String
Dim n As Integer
Dim StrLinea As String
Dim i As Long
Dim j As Long
Dim IntRenglones As Variant
Dim IntColumnas As Variant
Dim StrLineaEns As String


StrArchivo = App.Path & "\Templates\and4.esq"
n = FreeFile()
Open StrArchivo For Input As #n
  Line Input #n, IntRenglones
   For i = 1 To IntRenglones
      Line Input #n, IntColumnas
      For j = 1 To IntColumnas
         Line Input #n, StrLinea
         StrMatrizEsqueleto(i, j) = StrLinea
      Next j
   Next i
Close #n



StrMatrizEsqueleto(2, 4) = Mid(StrModulo, 8, 1)
StrMatrizEsqueleto(2, 6) = "gp" & LCase(Mid(StrModulo, 5, 3))
StrMatrizEsqueleto(2, 8) = "and4#00" & Mid(StrModulo, 3, 2) & " "
StrMatrizEsqueleto(4, 1) = "and4#00" & Mid(StrModulo, 3, 2) & ":       "
StrMatrizEsqueleto(4, 4) = Mid(StrModulo, 12, 1)

StrMatrizEsqueleto(4, 6) = "gp" & LCase(Mid(StrModulo, 9, 3))
StrMatrizEsqueleto(4, 8) = "and4#10" & Mid(StrModulo, 3, 2) & " "
StrMatrizEsqueleto(6, 1) = "and4#10" & Mid(StrModulo, 3, 2) & ":       "
StrMatrizEsqueleto(6, 4) = Mid(StrModulo, 16, 1)

StrMatrizEsqueleto(6, 6) = "gp" & LCase(Mid(StrModulo, 13, 3))
StrMatrizEsqueleto(6, 8) = "and4#20" & Mid(StrModulo, 3, 2) & " "
StrMatrizEsqueleto(8, 1) = "and4#20" & Mid(StrModulo, 3, 2) & ":       "
StrMatrizEsqueleto(8, 4) = Mid(StrModulo, 20, 1)

StrMatrizEsqueleto(8, 6) = "gp" & LCase(Mid(StrModulo, 17, 3))
StrMatrizEsqueleto(8, 8) = "and4#30" & Mid(StrModulo, 3, 2) & " "
StrMatrizEsqueleto(10, 1) = "and4#30" & Mid(StrModulo, 3, 2) & ":       "

If (Mid(StrModulo, 25, 4)) = 0 Then
   StrMatrizEsqueleto(11, 4) = "#$0f"
   End If
If (Mid(StrModulo, 25, 4)) = 1 Then
   StrMatrizEsqueleto(11, 4) = "#$0e"
   End If
If (Mid(StrModulo, 25, 4)) = 10 Then
   StrMatrizEsqueleto(11, 4) = "#$0d"
   End If
If (Mid(StrModulo, 25, 4)) = 11 Then
   StrMatrizEsqueleto(11, 4) = "#$0c"
End If

If (Mid(StrModulo, 25, 4)) = 100 Then
   StrMatrizEsqueleto(11, 4) = "#$0b"
   End If
If (Mid(StrModulo, 25, 4)) = 101 Then
   StrMatrizEsqueleto(11, 4) = "#$0a"
   End If
If (Mid(StrModulo, 25, 4)) = 110 Then
   StrMatrizEsqueleto(11, 4) = "#$09"
   End If
If (Mid(StrModulo, 25, 4)) = 111 Then
  StrMatrizEsqueleto(11, 4) = "#$08"
   End If
If (Mid(StrModulo, 25, 4)) = 1000 Then
   StrMatrizEsqueleto(11, 4) = "#$07"
   End If
If (Mid(StrModulo, 25, 4)) = 1001 Then
   StrMatrizEsqueleto(11, 4) = "#$06"
   End If
If (Mid(StrModulo, 25, 4)) = 1010 Then
   StrMatrizEsqueleto(11, 4) = "#$05"
   End If
If (Mid(StrModulo, 25, 4)) = 1011 Then
   StrMatrizEsqueleto(11, 4) = "#$04"
   End If
If (Mid(StrModulo, 25, 4)) = 1100 Then
   StrMatrizEsqueleto(11, 4) = "#$03"
   End If
If (Mid(StrModulo, 25, 4)) = 1101 Then
   StrMatrizEsqueleto(11, 4) = "#$02"
   End If
If (Mid(StrModulo, 25, 4)) = 1110 Then
   StrMatrizEsqueleto(11, 4) = "#$01"
   End If
If (Mid(StrModulo, 25, 4)) = 1111 Then
  StrMatrizEsqueleto(11, 4) = "#$00"
   End If

StrMatrizEsqueleto(13, 4) = "and4#40" & Mid(StrModulo, 3, 2) & " "
StrMatrizEsqueleto(14, 4) = Mid(StrModulo, 24, 1)
StrMatrizEsqueleto(14, 6) = "gp" & LCase(Mid(StrModulo, 21, 3))
StrMatrizEsqueleto(15, 4) = "and4#50" & Mid(StrModulo, 3, 2) & " "
StrMatrizEsqueleto(16, 1) = "and4#40" & Mid(StrModulo, 3, 2) & ":       "
StrMatrizEsqueleto(16, 4) = Mid(StrModulo, 24, 1)
StrMatrizEsqueleto(16, 6) = "gp" & LCase(Mid(StrModulo, 21, 3))
StrMatrizEsqueleto(17, 1) = "and4#50" & Mid(StrModulo, 3, 2) & ":       "


   For i = 1 To 17
      For j = 1 To 8
         StrLineaEns = StrLineaEns & StrMatrizEsqueleto(i, j)
      Next j
      StrLineaEns = StrLineaEns & vbCrLf
   Next i

   StrEnsResultadoMatriz = StrLineaEns

End Sub
```

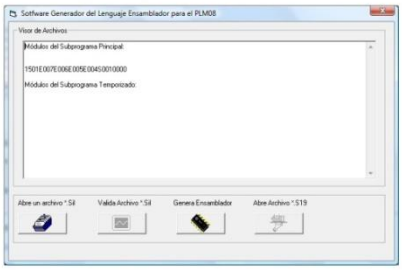


Figura 4.2.8 Código en Visual Basic ® módulo AND 4 entradas y ventana que muestra la cadena asociada StrModulo al validar un archivo.

4.2.6 Esqueleto de programación y matriz de cadenas asociadas con el módulo lógico que realiza temporizadores monodisparo (TEMPOC).

En la sección 3.2.6 del capítulo anterior se mostró el programa en lenguaje ensamblador asociado con el módulo lógico que realiza temporizador monodisparo (TEMPOC). El programa ensamblador genérico se transcribe a continuación:

```
                lda tesppspt
                bne npptc#nm
resettc#nm:     bclr bitsal,grupobitsal
inicontc#nm:    mov #$FF,dirbtempos+nn
                ldhx #$FFFF
                sthx dirbtempos+nnn
                bra salidatc#nm
npptc#nm:       brclr bitrst,grupobitrst,resettc#nm
                lda grupoedis+1
                and #$XX
                sta locaux#1
                lda grupoedis
                and #$YY
                cmp locaux#1
                beq siguetc3#nm
                bhi versaltc#nm
siguetc3#nm:    brclr bitsal,grupobitsal,salidatc#nm
                ldhx dirbtempos+nnn
                aix #$ff
                sthx dirbtempos+nnn
                cphx #$0000
                bne salidatc#nm
                lda dirbtempos+nn
                beq resettc#nm
                dec dirbtempos+nn
salidatc#nm:    bra sigblotc#nm
versaltc#nm:    bset bitsal,grupobitsal
                bra inicontc#nm
sigblotc#nm:    nop
```

El esqueleto de programación puede ser construido con un arreglo matricial, donde los distintos elementos (i, j) de la matriz corresponden en posición y orden a los caracteres propios del programa genérico que resuelve cada módulo lógico realizable.

	1	2	3	4	5	6	7	8
1		lda		tesppspt				
2		bne		npptc#**nm**				
3	resettc#**nm**:	bclr		**bitsal**	,	**grupobitsal**		
4	inicontc#**nm**:	mov		#$**FF**	,	dirbtempos+**nn**		
5		ldhx		#$**FFFF**				
6		sthx		dirbtempos+**nnn**				
7		bra		salidatc#**nm**				
8	npptc#**nm**:	brclr		**bitrst**	,	**grupobitrst**	,	resettc#**nm**
9		lda		**grupoedis**+1				
10		and		#$**XX**				
11		sta		locaux#1				
12		lda		**grupoedis**				
13		and		#$**YY**				
14		cmp		locaux#1				
15		beq		siguetc#3**nm**				
16		bhi		versaltc#**nm**				
17	siguetc#3**nm**:	brclr		**bitsal**	,	**grupobitsal**	,	salidatc#**nm**
18		ldhx		dirbtempos+**nnn**				
19		aix		#$ff				
20		sthx		dirbtempos+**nnn**				
21		cphx		#$0000				
22		bne		salidatc#**nm**				
23		lda		dirbtempos+**nn**				
24		beq		resettc#**nm**				
25		dec		dirbtempos+**nn**				
26	salidatc#**nm**:	bra		sigblotc#**nm**				
27	versaltc#**nm**:	bset		**bitsal**	,	**grupobitsal**		
28		bra		inicontc#**nm**				
29	sigblotc#**nm**:	nop						

Tabla 4.6 Matriz de cadenas asociada con el módulo lógico one-shot (TEMPOC).

Después el esqueleto de programación del módulo lógico realizable por él PLM08 será llamado por el programa GEN_ENS_PLM08 para hacer una sustitución de valores acorde entre los datos ingresados por el usuario al programar en lenguaje SIIL1 y las cadenas mudas vistas en la sección 3.2.6 del capítulo anterior. Generando con ello el programa en lenguaje ensamblador buscado para este caso.

Se hace una sustitución en la matriz de cadenas asociadas, denominada para el caso del código fuente del programa GEN_ENS_PLM08 como StrMatrizEsqueleto. Y una cadena denominada StrModulo. La cadena asociada StrModulo se genera al validar un archivo SIIL1 con el programa.

GEN_ENS_PLM08; esto para cada módulo lógico declarado por el usuario, (ver sección 4.3 de este capítulo, guía rápida del usuario).

La cadena StrModulo contiene información básica de las cadenas mudas a sustituir, para generar el archivo en código ensamblador para cada módulo lógico.

Por ejemplo; para la siguiente sentencia en SIIL1 que representa un temporizador monodisparo de tipo dos, de manera que las entradas de disparo y restablecimiento sean respectivamente las entradas E07 y E06, y la salida T sea S01; el pulso de salida es verificado en bajo y tiene una duración de 10 segundos; el disparo debe ser por flanco de bajada y el restablecimiento debe ser por nivel alto.

La sentencia SIIL1 correspondiente es la siguiente:

TEMPOC#2 E07,E06,S01,00:00:10.00,000;

Para el caso de este módulo lógico, la cadena StrModulo es:

2502E007E006S00100:00:10.00000

Para la cual, el primer par de dígitos representa un código numérico, que indica se trata de un módulo lógico temporizador monodisparo tipo dos (TEMPOC); el siguiente par de dígitos representa el número que el usuario asigno a este temporizador; los siguientes tres caracteres indican que la entrada de disparo “D” del temporizador pertenece al grupo cero de entradas, el siguiente carácter indica que el bit asociado con la entrada “D” es el bit 7; los siguientes tres caracteres indican que la entrada de restablecimiento “R” del temporizador pertenece al grupo cero de entradas, el siguiente carácter indica que el bit asociado con la entrada “R” es el bit 6. Los siguientes tres caracteres indican que la salida “T" pertenece al grupo cero de salidas y el siguiente carácter indica que el bit físico de salidas, es el bit 1 del grupo antes mencionado. Los siguientes once caracteres indican la duración del pulso de salida, para este caso 10 segundos. Y los últimos tres dígitos indican que el temporizador se dispara por flanco de bajada; es restablecido por nivel alto y el nivel de verificación de la salida es bajo, respectivamente.

A continuación se muestra en la figura 4.2.9; el código en Visual Basic® que hace la asignación de cadenas mudas en la matriz correspondiente para el caso de un módulo lógico temporizador monodisparo de tipo 2 (TEMPOC).

```
Public Sub TempoC(StrModulo As String)

Dim StrArchivo As String
Dim n As Integer
Dim StrLinea As String
Dim i As Long
Dim j As Long
Dim IntRenglones As Variant
Dim IntColumnas As Variant
Dim StrLineaEns As String
Dim Residuo As Integer
Dim HexaStr As String
Dim D As Long

ReDim StrMatrizEsqueleto(30, 21)

StrArchivo = App.Path & "\Templates\tempoc.esq"
n = FreeFile()
Open StrArchivo For Input As #n
  Line Input #n, IntRenglones
   For i = 1 To IntRenglones
      Line Input #n, IntColumnas
      For j = 1 To IntColumnas
         Line Input #n, StrLinea
         StrMatrizEsqueleto(i, j) = StrLinea
      Next j
   Next i
Close #n


   'RUTINA RELOJ

nn = (Mid(StrModulo, 3, 2) - 1) * 3
 nnn = nn + 1


If (Mid(StrModulo, 28, 3)) = 0 Then        '000
   StrMatrizEsqueleto(3, 2) = "bset"
   StrMatrizEsqueleto(8, 2) = "brset"
   StrMatrizEsqueleto(16, 2) = "blo"
   StrMatrizEsqueleto(17, 2) = "brset"
   StrMatrizEsqueleto(27, 2) = "bclr"
   End If
If (Mid(StrModulo, 28, 3)) = 1 Then        '001
   StrMatrizEsqueleto(3, 2) = "bclr"
   StrMatrizEsqueleto(8, 2) = "brset"
   StrMatrizEsqueleto(16, 2) = "blo"
   StrMatrizEsqueleto(17, 2) = "brclr"
   StrMatrizEsqueleto(27, 2) = "bset"
   End If
If (Mid(StrModulo, 28, 3)) = 10 Then       '010
   StrMatrizEsqueleto(3, 2) = "bset"
   StrMatrizEsqueleto(8, 2) = "brclr"
   StrMatrizEsqueleto(16, 2) = "blo"
   StrMatrizEsqueleto(17, 2) = "brset"
   StrMatrizEsqueleto(27, 2) = "bclr"
   End If
If (Mid(StrModulo, 28, 3)) = 11 Then       '011
   StrMatrizEsqueleto(3, 2) = "bclr"
   StrMatrizEsqueleto(8, 2) = "brclr"
   StrMatrizEsqueleto(16, 2) = "blo"
   StrMatrizEsqueleto(17, 2) = "brclr"
   StrMatrizEsqueleto(27, 2) = "bset"
End If

    .
    .
    .

 If (Mid(StrModulo, 28, 3)) = 111 Then     '111
   StrMatrizEsqueleto(3, 2) = "bclr"
   StrMatrizEsqueleto(8, 2) = "brclr"
   StrMatrizEsqueleto(16, 2) = "bhi"
   StrMatrizEsqueleto(17, 2) = "brclr"
   StrMatrizEsqueleto(27, 2) = "bset"
   End If
```

```
StrMatrizEsqueleto(3, 4) = LCase(Mid(StrModulo, 16, 1))
StrMatrizEsqueleto(3, 6) = "gp" & LCase(Mid(StrModulo, 13, 3))
StrMatrizEsqueleto(4, 4) = "#$" & E3
StrMatrizEsqueleto(5, 4) = "#$" & E8
StrMatrizEsqueleto(4, 6) = "dirbtempos+" & nn
StrMatrizEsqueleto(6, 4) = "dirbtempos+" & nnn

StrMatrizEsqueleto(8, 4) = LCase(Mid(StrModulo, 12, 1))
StrMatrizEsqueleto(8, 6) = "gp" & LCase(Mid(StrModulo, 9, 3))
StrMatrizEsqueleto(9, 4) = "gp" & LCase(Mid(StrModulo, 5, 3)) & "+1"
StrMatrizEsqueleto(10, 4) = "#$" & Y
StrMatrizEsqueleto(12, 4) = "gp" & LCase(Mid(StrModulo, 5, 3))
StrMatrizEsqueleto(13, 4) = "#$" & Y

StrMatrizEsqueleto(17, 4) = LCase(Mid(StrModulo, 16, 1))
StrMatrizEsqueleto(17, 6) = "gp" & LCase(Mid(StrModulo, 13, 3))
StrMatrizEsqueleto(18, 4) = "dirbtempos+" & nnn
StrMatrizEsqueleto(20, 4) = "dirbtempos+" & nnn
StrMatrizEsqueleto(23, 4) = "dirbtempos+" & nn
                                  'StrMatrizEsqueleto(24, 4) = "dirbtempos+" & nnn
StrMatrizEsqueleto(25, 4) = "dirbtempos+" & nn
StrMatrizEsqueleto(27, 4) = LCase(Mid(StrModulo, 16, 1))
StrMatrizEsqueleto(27, 6) = "gp" & LCase(Mid(StrModulo, 13, 3))

StrMatrizEsqueleto(3, 1) = "resettc#" & Mid(StrModulo, 3, 2) & ":       "
StrMatrizEsqueleto(4, 1) = "inicontc#" & Mid(StrModulo, 3, 2) & ":      "
StrMatrizEsqueleto(8, 1) = "npptc#" & Mid(StrModulo, 3, 2) & ":         "
StrMatrizEsqueleto(17, 1) = "siguetc3#" & Mid(StrModulo, 3, 2) & ":       "
StrMatrizEsqueleto(26, 1) = "salidatc#" & Mid(StrModulo, 3, 2) & ":       "
StrMatrizEsqueleto(27, 1) = "versaltc#" & Mid(StrModulo, 3, 2) & ":       "
StrMatrizEsqueleto(29, 1) = "sigblotc" & Mid(StrModulo, 3, 2) & ":       "

StrMatrizEsqueleto(2, 4) = "npptc#" & Mid(StrModulo, 3, 2) & "        "
StrMatrizEsqueleto(7, 4) = "salidatc#" & Mid(StrModulo, 3, 2) & "       "
StrMatrizEsqueleto(8, 8) = "resettc#" & Mid(StrModulo, 3, 2) & "       "
StrMatrizEsqueleto(15, 4) = "siguetc3#" & Mid(StrModulo, 3, 2) & "      "
StrMatrizEsqueleto(16, 4) = "versaltc#" & Mid(StrModulo, 3, 2) & "      "
StrMatrizEsqueleto(17, 8) = "salidatc#" & Mid(StrModulo, 3, 2) & "      "
StrMatrizEsqueleto(22, 4) = "salidatc#" & Mid(StrModulo, 3, 2) & "      "
StrMatrizEsqueleto(24, 4) = "resettc#" & Mid(StrModulo, 3, 2) & "       "
StrMatrizEsqueleto(26, 4) = "sigblotc" & Mid(StrModulo, 3, 2) & "       "
StrMatrizEsqueleto(28, 4) = "inicontc#" & Mid(StrModulo, 3, 2) & "      "


 For i = 1 To 30
     For j = 1 To 8
        StrLineaEns = StrLineaEns & StrMatrizEsqueleto(i, j)
     Next j
     StrLineaEns = StrLineaEns & vbCrLf
  Next i

  StrEnsResultadoMatriz = StrLineaEns


End Sub
```

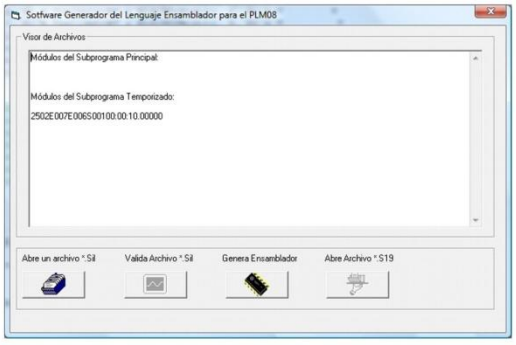


Figura 4.2.9 Código en Visual Basic ® módulo temporizador monodisparo (TEMPOC) y ventana que muestra la cadena asociada StrModulo al validar un archivo.

4.2.7 Esqueleto de programación y matriz de cadenas asociadas con el módulo lógico que realiza temporizadores con retardo a la activación (On Delay) y con retardo a la desactivación (Off Delay) (TEMPOD).

En la sección 3.2.7 del capítulo anterior se mostró el programa en lenguaje ensamblador asociado con el módulo lógico que realiza temporizador con retardo a la activación o retardo a la desactivación. (TEMPOD).

El programa ensamblador genérico se transcribe a continuación:

```
                lda tesppspt
                bne npptc#nm
resettc#nm:     bclr bitsal,grupobitsal
inicontc#nm:    mov #$FF,dirbtempos+nn
                ldhx #$FFFF
                sthx dirbtempos+nnn
                bra salidatc#nm
npptc#nm:       brclr bitdsp,grupobitdsp,resettc#nm
                brclr bitrest,grupobitrest,resettc#nm
                brset bitsal,grupobitsal,salidatc#nm
                ldhx dirbtempos+nnn
                aix #$FF
                sthx dirbtempos+nnn
                ldhx dirbtempos+nnn
                cphx #$FFFF
                beq versaltc#nm
                bne salidatc#nm
                dec dirbtempos+nn
                ldhx dirbtempos+nnn
                cphx #$0000
                bne salidatc#nm
                lda dirbtempos+nn
                beq versaltc#nm
salidatc#nm:    bra sigblotc#nm
versaltc#nm:    bset bitsal,grupobitsal
sigblotc#nm:    nop
```

El esqueleto de programación puede ser construido con un arreglo matricial, donde los distintos elementos (i, j) de la matriz corresponden en posición y orden a los caracteres propios del programa genérico que resuelve cada módulo lógico realizable.

	1	2	3	4	5	6	7	8
1		lda		tesppspt				
2		bne		npptc#**nm**				
3	resettc#**nm**:	bclr		**bitsal**	,	**grupobitsal**		
4	inicontc#**nm**:	mov		#$**FF**	,	dirbtempos+**nn**		
5		ldhx		#$**FFFF**				
6		sthx		dirbtempos+**nnn**				
7		bra		salidatc#**nm**				
8	npptc#**nm**:	brclr		**bitdsp**	,	**grupobitdsp**	,	resettc#**nm**
9		brclr		**bitrest**	,	**grupobitrest**	,	resettc#**nm**
10		brset		bitsal	,	**grupobitsal**	,	salidatc#**nm**
11		ldhx		dirbtempos+**nnn**				
12		aix		**#$FF**				
13		sthx		dirbtempos+**nnn**				
14		ldhx		dirbtempos+**nnn**				
15		cphx		#$**FFFF**				
16		beq		**versaltc#nm**				
17		bne		**salidatc#nm**				
18		dec		dirbtempos+**nn**				
19		ldhx		dirbtempos+**nnn**				
20		cphx		#$**0000**				
21		bne		salidatc#**nm**				
22		lda		dirbtempos+**nn**				
23		beq		versaltc#**nm**				
24	salidatc#**nm**:	bra		sigblotc#**nm**				
25	versaltc#**nm**:	bset		**bitsal**	,	**grupobitsal**		
26	sigblotc#**nm**:	nop						

Tabla 4.7 Matriz de cadenas asociada con el módulo temporizador con retardo a la activación ó retardo a la desactivación (TEMPOD).

Después el esqueleto de programación del módulo lógico realizable por él PLM08 será llamado por el programa GEN_ENS_PLM08 para hacer una sustitución de valores acorde entre los datos ingresados por el usuario al programar en lenguaje SIIL1 y las cadenas mudas vistas en la sección 3.2.7 del capítulo anterior. Generando con ello el programa en lenguaje ensamblador buscado para este caso.

Se hace una sustitución en la matriz de cadenas asociadas, denominada para el caso del código fuente del programa GEN_ENS_PLM08 como StrMatrizEsqueleto. Y una cadena denominada StrModulo. La cadena asociada StrModulo se genera al validar un archivo SIIL1 con el programa GEN_ENS_PLM08; esto para cada módulo lógico declarado por el usuario, (ver sección 4.3 de este capítulo, guía rápida del usuario).

La cadena StrModulo contiene información básica de las cadenas mudas a sustituir, para generar el archivo en código ensamblador para cada módulo lógico.

Por ejemplo; para la siguiente sentencia en SIIL1 que representa un temporizador con retardo a la desactivación de 10 segundos, con restablecimiento por nivel alto, asignándosele el número de temporizador 3; las entradas de disparo y restablecimiento son E07 y E06, y la salida es S01.

La sentencia SIIL1 correspondiente es la siguiente:

TEMPOD#3 E07,E06,S01,00:00:10.00,00;

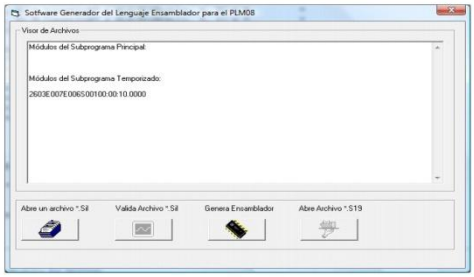
Para el caso de este módulo lógico, la cadena StrModulo es:

2603E007E006S00100:00:10.0000

Para la cual, el primer par de dígitos representa un código numérico, que indica se trata de un módulo lógico temporizador con retardo a la activación o desactivación (TEMPOD); el siguiente par de dígitos representa el número que el usuario asigno a este temporizador; los siguientes tres caracteres indican que la entrada de disparo “D” del temporizador pertenece al grupo cero de entradas, el siguiente carácter indica que el bit asociado con la entrada “D” es el bit 7; los siguientes tres caracteres indican que la entrada de restablecimiento “R” del temporizador pertenece al grupo cero de entradas, el siguiente carácter indica que el bit asociado con la entrada “R” es el bit 6. Los siguientes tres caracteres indican que la salida pertenece al grupo cero de salidas y el siguiente carácter indica que el bit físico de salidas, es el bit 1 del grupo antes mencionado. Los siguientes once caracteres indican la duración del pulso de salida, para este caso 10 segundos. Y los últimos dos dígitos indican que el temporizador presenta retardo a la desactivación y es restablecido por nivel alto, respectivamente.

A continuación se muestra en la figura 4.2.10; el código en Visual Basic® que hace la asignación de cadenas mudas en la matriz correspondiente para el caso de un módulo lógico temporizador con retardo a la activación (on delay) o con retardo a la desactivación (off delay). (TEMPOD).

```
Public Sub TempoD(StrModulo As String)

Dim StrArchivo As String
Dim n As Integer
Dim StrLinea As String
Dim i As Long
Dim j As Long
Dim IntRenglones As Variant
Dim IntColumnas As Variant
Dim StrLineaEns As String
Dim Residuo As Integer
Dim HexaStr As String
Dim D As Long

ReDim StrMatrizEsqueleto(28, 21)

StrArchivo = App.Path & "\Templates\tempod.esq"
n = FreeFile()
Open StrArchivo For Input As #n
  Line Input #n, IntRenglones
  For i = 1 To IntRenglones
     Line Input #n, IntColumnas
     For j = 1 To IntColumnas
        Line Input #n, StrLinea
        StrMatrizEsqueleto(i, j) = StrLinea
     Next j
  Next i
Close #n
                'RUTINA RELOJ

nn = (Mid(StrModulo, 3, 2) - 1) * 3
 nnn = nn + 1


If (Mid(StrModulo, 28, 2)) = 0 Then
   StrMatrizEsqueleto(3, 2) = "bset"
   StrMatrizEsqueleto(8, 2) = "brclr"
   StrMatrizEsqueleto(9, 2) = "brset"
   StrMatrizEsqueleto(10, 2) = "brclr"
   StrMatrizEsqueleto(25, 2) = "bclr"
   End If
If (Mid(StrModulo, 28, 2)) = 1 Then
   StrMatrizEsqueleto(3, 2) = "bset"
   StrMatrizEsqueleto(8, 2) = "brclr"
   StrMatrizEsqueleto(9, 2) = "brclr"
   StrMatrizEsqueleto(10, 2) = "brclr"
   StrMatrizEsqueleto(25, 2) = "bclr"
   End If
If (Mid(StrModulo, 28, 2)) = 10 Then
   StrMatrizEsqueleto(3, 2) = "bclr"
   StrMatrizEsqueleto(8, 2) = "brclr"
   StrMatrizEsqueleto(9, 2) = "brset"
   StrMatrizEsqueleto(10, 2) = "brset"
   StrMatrizEsqueleto(25, 2) = "bset"
   End If
If (Mid(StrModulo, 28, 2)) = 11 Then
   StrMatrizEsqueleto(3, 2) = "bclr"
   StrMatrizEsqueleto(8, 2) = "brclr"
   StrMatrizEsqueleto(9, 2) = "brclr"
   StrMatrizEsqueleto(10, 2) = "brset"
   StrMatrizEsqueleto(25, 2) = "bset"
End If

StrMatrizEsqueleto(2, 4) = "npptc#" & Mid(StrModulo, 3, 2) & "       "
StrMatrizEsqueleto(3, 1) = "resettc#" & Mid(StrModulo, 3, 2) & ":       "
StrMatrizEsqueleto(4, 1) = "inicontc#" & Mid(StrModulo, 3, 2) & ":      "
StrMatrizEsqueleto(7, 4) = "salidatc#" & Mid(StrModulo, 3, 2) & "       "
StrMatrizEsqueleto(8, 1) = "npptc#" & Mid(StrModulo, 3, 2) & ":       "
StrMatrizEsqueleto(8, 8) = "resettc#" & Mid(StrModulo, 3, 2) & "       "
StrMatrizEsqueleto(9, 8) = "resettc#" & Mid(StrModulo, 3, 2) & "       "
StrMatrizEsqueleto(10, 8) = "salidatc#" & Mid(StrModulo, 3, 2) & "       "

StrMatrizEsqueleto(16, 4) = "versaltc#" & Mid(StrModulo, 3, 2) & "       "
StrMatrizEsqueleto(17, 4) = "salidatc#" & Mid(StrModulo, 3, 2) & "       "
StrMatrizEsqueleto(21, 4) = "salidatc#" & Mid(StrModulo, 3, 2) & "       "
StrMatrizEsqueleto(23, 4) = "versaltc#" & Mid(StrModulo, 3, 2) & "       "
StrMatrizEsqueleto(24, 1) = "salidatc#" & Mid(StrModulo, 3, 2) & ":      "
StrMatrizEsqueleto(24, 4) = "sigblotc#" & Mid(StrModulo, 3, 2) & "       "
StrMatrizEsqueleto(25, 1) = "versaltc#" & Mid(StrModulo, 3, 2) & ":      "
StrMatrizEsqueleto(26, 1) = "sigblotc#" & Mid(StrModulo, 3, 2) & ":      "


  For i = 1 To 28
     For j = 1 To 8
        StrLineaEns = StrLineaEns & StrMatrizEsqueleto(i, j)
     Next j
     StrLineaEns = StrLineaEns & vbCrLf
  Next i

  StrEnsResultadoMatriz = StrLineaEns

End Sub
```




```
StrMatrizEsqueleto(3, 4) = LCase(Mid(StrModulo, 16, 1))
StrMatrizEsqueleto(3, 6) = "gp" & LCase(Mid(StrModulo, 13, 3))
StrMatrizEsqueleto(4, 4) = "#$" & E3
StrMatrizEsqueleto(5, 4) = "#$" & E8
StrMatrizEsqueleto(4, 6) = "dirbtempos+" & nn
StrMatrizEsqueleto(6, 4) = "dirbtempos+" & nnn
StrMatrizEsqueleto(8, 4) = LCase(Mid(StrModulo, 8, 1))
StrMatrizEsqueleto(8, 6) = "gp" & LCase(Mid(StrModulo, 5, 3))
StrMatrizEsqueleto(9, 4) = LCase(Mid(StrModulo, 12, 1))
StrMatrizEsqueleto(9, 6) = "gp" & LCase(Mid(StrModulo, 9, 3))

StrMatrizEsqueleto(10, 4) = LCase(Mid(StrModulo, 16, 1))
StrMatrizEsqueleto(10, 6) = "gp" & LCase(Mid(StrModulo, 13, 3))
StrMatrizEsqueleto(11, 4) = "dirbtempos+" & nnn
StrMatrizEsqueleto(13, 4) = "dirbtempos+" & nnn
StrMatrizEsqueleto(14, 4) = "dirbtempos+" & nnn
StrMatrizEsqueleto(18, 4) = "dirbtempos+" & nn
StrMatrizEsqueleto(19, 4) = "dirbtempos+" & nnn

StrMatrizEsqueleto(22, 4) = "dirbtempos+" & nn
StrMatrizEsqueleto(25, 4) = LCase(Mid(StrModulo, 16, 1))
StrMatrizEsqueleto(25, 6) = "gp" & LCase(Mid(StrModulo, 13, 3))
```

Figura 4.2.10 Código en Visual Basic ® módulo temporizador con retardo a la activación ó desactivación (TEMPOD) y ventana que muestra la cadena asociada StrModulo al validar un archivo.

4.2.8 Esqueleto de programación y matriz de cadenas asociadas con el módulo lógico que realiza temporizadores astables (TEMPOE).

En la sección 3.2.8 del capítulo anterior se mostró el programa en lenguaje ensamblador asociado con el módulo lógico que realiza temporizador de tipo astable.

El programa ensamblador genérico se transcribe a continuación:

```
                lda tesppspt
                bne npptc#0
resettc#0:      bclr bitsal,grupobitsal
etiqueta1:      bset bitsal,grupobitsal
inicontc#0:     mov #$FF,dirbtempos+nn
                ldhx #$FFFF
                sthx dirbtempos+nnn
                bra salidatc#0
npptc#0:        brclr bitrst,grupobitrst,resettc#0
                lda dirbtempos+nn
                cmp #$00
                bne decrementa#0
                ldhx dirbtempos+nnn
                cphx #$FFFF
                bne decrementa#0
                bclr bitsal,grupobitsal
decrementa#0:   ldhx dirbtempos+nnn
                aix #$FF
                sthx dirbtempos+nnn
                ldhx dirbtempos+nnn
                cphx #$FFFF
                beq etiqueta1
                bne salidatc#0
                dec dirbtempos+nn
                ldhx dirbtempos+nnn
                cphx #$0000
                bne salidatc#0
                lda dirbtempos+nn
                beq etiqueta1
salidatc#0:     bra sigblotc#0
sigblotc#0:     nop
```

El esqueleto de programación puede ser construido con un arreglo matricial, donde los distintos elementos (i, j) de la matriz corresponden en posición y orden a los caracteres propios del programa genérico que resuelve cada módulo lógico realizable.

	1	2	3	4	5	6	7	8
1		lda		tesppspt				
2		bne		npptc**#nm**				
3	resettc**#nm**:	bclr		**bitsal**	,	**grupobitsal**		
4	etiqueta**nm**:	bset		**bitsal**	,	**grupobitsal**		
5	inicontc**#nm**:	mov		#$**FF**	,	dirbtempos+**nn**		
6		Ldhx		#$**FFFF**				
7		sthx		dirbtempos+**nnn**				
8		bra		**salidatc#nm**				
9	npptc**#nm**:	brclr		**bitrest**	,	**grupobitrest**	,	resettc**#nm**
10		lda		dirbtempos+**nn**				
11		cmp		#$00				
12		bne		**decrementa#nm**				
13		ldhx		dirbtempos+**nnn**				
14		cphx		#$**FFFF**				
15		bne		decrementa**#nm**				
16		bclr		**bitsal**	,	**grupobitsal**		
17	decrementa**#nm**:	ldhx		dirbtempos+**nnn**				
18		aix		#$**FF**				
19		sthx		dirbtempos+**nnn**				
20		ldhx		dirbtempos+**nnn**				
21		cphx		#$**AAAA**				
22		beq		etiqueta**nm**				
23		bne		salidatc**#nm**				
24		dec		dirbtempos+**nn**				
25		ldhx		dirbtempos+**nnn**				
26		cphx		#$0000				
27		bne		salidatc**#nm**				
28		lda		dirbtempos+**nn**				
29		beq		etiqueta**nm**				
30	salidatc**#nm**	bra		sigblotc**#nm**				
31	sigblotc**#nm**	nop						

Tabla 4.8 Matriz de cadenas asociada con el módulo temporizador astable (TEMPOE).

Después el esqueleto de programación del módulo lógico realizable por él PLM08 será llamado por el programa GEN_ENS_PLM08 para hacer una sustitución de valores acorde entre los datos ingresados por el usuario al programar en lenguaje SIIL1 y las cadenas mudas vistas en la sección

3.2.8 del capítulo anterior. Generando con ello el programa en lenguaje ensamblador buscado para este caso.

Se hace una sustitución en la matriz de cadenas asociadas, denominada para el caso del código fuente del programa GEN_ENS_PLM08 como StrMatrizEsqueleto. Y una cadena denominada StrModulo. La cadena asociada StrModulo se genera al validar un archivo SIIL1 con el programa GEN_ENS_PLM08; esto para cada módulo lógico declarado por el usuario, (ver sección 4.3 de este capítulo, guía rápida del usuario).

La cadena StrModulo contiene información básica de las cadenas mudas a sustituir, para generar el archivo en código ensamblador para cada módulo lógico.

Por ejemplo; para la siguiente sentencia en SIIL1 que representa un temporizador astable que arranca en cero; y los tiempos Tm y Tc sean respectivamente 2 segundos y 1 segundo y que su nivel de restablecimiento sea por nivel alto.

La sentencia SIIL1 correspondiente es la siguiente:

TEMPOE#5 E07,S00,00:00:00.02,00:00:00.01,00;

Para el caso de este módulo lógico, la cadena StrModulo es:

2705E007S00000:00:02.0000:00:01.0000

Para la cual, el primer par de dígitos representa un código numérico, que indica se trata de un módulo lógico temporizador astable (TEMPOE); el siguiente par de dígitos representa el número que el usuario asigno a este temporizador; los siguientes tres caracteres indican que la entrada de restablecimiento “R” del temporizador pertenece al grupo cero de entradas, el siguiente carácter indica que el bit asociado con la entrada “R” es el bit 7. Los siguientes tres caracteres indican que la salida “T” pertenece al grupo cero de salidas y el siguiente carácter indica que el bit físico de salidas, es el bit 0 del grupo antes mencionado. Los siguientes once caracteres indican la duración del pulso Tm, para este caso 2 segundos. Y los siguientes once caracteres indican la duración del pulso Tc, para este caso 1 segundo. Los últimos dos dígitos indican que el temporizador se restablece por nivel alto y arranca en cero, respectivamente.

A continuación se muestra en la figura 4.2.11; el código en Visual Basic® que hace la asignación de cadenas mudas en la matriz correspondiente para el caso de un módulo lógico temporizador astable. (TEMPOE).

```
Public Sub TempoE(StrModulo As String)

Dim StrArchivo As String
Dim n As Integer
Dim StrLinea As String
Dim i As Long
Dim j As Long
Dim IntRenglones As Variant
Dim IntColumnas As Variant
Dim StrLineaEns As String
Dim Residuo As Integer
Dim HexaStr As String
Dim D As Long


ReDim StrMatrizEsqueleto(33, 21)

StrArchivo = App.Path & "\Templates\tempoe.esq"
n = FreeFile()
Open StrArchivo For Input As #n
  Line Input #n, IntRenglones
   For i = 1 To IntRenglones
      Line Input #n, IntColumnas
      For j = 1 To IntColumnas
         Line Input #n, StrLinea
         StrMatrizEsqueleto(i, j) = StrLinea
      Next j
   Next i
Close #n

                'RUTINA RELOJ 1

                'RUTINA RELOJ 2

n = (Mid(StrModulo, 3, 2) - 1) * 3
 nnn = nn + 1


If (Mid(StrModulo, 35, 2)) = 0 Then
   StrMatrizEsqueleto(3, 2) = "bclr"
   StrMatrizEsqueleto(4, 2) = "bclr"
   StrMatrizEsqueleto(9, 2) = "brset"
   StrMatrizEsqueleto(16, 2) = "bset"
'   StrMatrizEsqueleto(30, 2) = "bclr"

   End If

If (Mid(StrModulo, 35, 2)) = 1 Then
   StrMatrizEsqueleto(3, 2) = "bclr"
   StrMatrizEsqueleto(4, 2) = "bset"
   StrMatrizEsqueleto(9, 2) = "brset"
   StrMatrizEsqueleto(16, 2) = "bclr"
'   StrMatrizEsqueleto(30, 2) = "bset"
   End If

If (Mid(StrModulo, 35, 2)) = 10 Then
   StrMatrizEsqueleto(3, 2) = "bclr"
   StrMatrizEsqueleto(4, 2) = "bclr"
   StrMatrizEsqueleto(9, 2) = "brclr"
   StrMatrizEsqueleto(16, 2) = "bset"
'   StrMatrizEsqueleto(30, 2) = "bclr"
   End If

If (Mid(StrModulo, 35, 2)) = 11 Then
   StrMatrizEsqueleto(3, 2) = "bclr"
   StrMatrizEsqueleto(4, 2) = "bset"
   StrMatrizEsqueleto(9, 2) = "brclr"
   StrMatrizEsqueleto(16, 2) = "bclr"
'   StrMatrizEsqueleto(30, 2) = "bset"
End If
```

```
StrMatrizEsqueleto(3, 4) = LCase(Mid(StrModulo, 12, 1))
StrMatrizEsqueleto(3, 6) = "gp" & LCase(Mid(StrModulo, 9, 3))
StrMatrizEsqueleto(4, 4) = LCase(Mid(StrModulo, 12, 1))
StrMatrizEsqueleto(4, 6) = "gp" & LCase(Mid(StrModulo, 9, 3))
StrMatrizEsqueleto(5, 4) = "#$" & E3
StrMatrizEsqueleto(6, 4) = "#$" & E8
StrMatrizEsqueleto(5, 6) = "dirbtempos+" & nn
StrMatrizEsqueleto(7, 4) = "dirbtempos+" & nnn
StrMatrizEsqueleto(9, 4) = LCase(Mid(StrModulo, 8, 1))
StrMatrizEsqueleto(9, 6) = "gp" & LCase(Mid(StrModulo, 5, 3))
StrMatrizEsqueleto(10, 4) = "dirbtempos+" & nn
StrMatrizEsqueleto(13, 4) = "dirbtempos+" & nnn
StrMatrizEsqueleto(11, 4) = "#$" & C3
StrMatrizEsqueleto(14, 4) = "#$" & C8

StrMatrizEsqueleto(16, 4) = LCase(Mid(StrModulo, 12, 1))
StrMatrizEsqueleto(16, 6) = "gp" & LCase(Mid(StrModulo, 9, 3))
StrMatrizEsqueleto(17, 4) = "dirbtempos+" & nnn
StrMatrizEsqueleto(19, 4) = "dirbtempos+" & nnn
StrMatrizEsqueleto(20, 4) = "dirbtempos+" & nnn
StrMatrizEsqueleto(24, 4) = "dirbtempos+" & nn
StrMatrizEsqueleto(25, 4) = "dirbtempos+" & nnn

StrMatrizEsqueleto(28, 4) = "dirbtempos+" & nn

StrMatrizEsqueleto(2, 4) = "npptc#" & Mid(StrModulo, 3, 2) & "        "
StrMatrizEsqueleto(3, 1) = "resettc#" & Mid(StrModulo, 3, 2) & ":       "
StrMatrizEsqueleto(4, 1) = "etiqueta#" & Mid(StrModulo, 3, 2) & ":       "
StrMatrizEsqueleto(5, 1) = "inicontc#" & Mid(StrModulo, 3, 2) & ":       "
StrMatrizEsqueleto(8, 4) = "salidatc#" & Mid(StrModulo, 3, 2) & "       "
StrMatrizEsqueleto(9, 1) = "npptc#" & Mid(StrModulo, 3, 2) & ":       "
StrMatrizEsqueleto(9, 8) = "resettc#" & Mid(StrModulo, 3, 2) & "       "
StrMatrizEsqueleto(12, 4) = "decrementa#" & Mid(StrModulo, 3, 2) & "       "

StrMatrizEsqueleto(15, 4) = "decrementa#" & Mid(StrModulo, 3, 2) & "       "
StrMatrizEsqueleto(17, 1) = "decrementa#" & Mid(StrModulo, 3, 2) & ":       "
StrMatrizEsqueleto(22, 4) = "etiqueta#" & Mid(StrModulo, 3, 2) & "       "
StrMatrizEsqueleto(23, 4) = "salidatc#" & Mid(StrModulo, 3, 2) & "       "
StrMatrizEsqueleto(27, 4) = "salidatc#" & Mid(StrModulo, 3, 2) & "       "
StrMatrizEsqueleto(29, 4) = "etiqueta#" & Mid(StrModulo, 3, 2) & "       "
StrMatrizEsqueleto(30, 1) = "salidatc#" & Mid(StrModulo, 3, 2) & ":       "
StrMatrizEsqueleto(30, 4) = "sigblotc#" & Mid(StrModulo, 3, 2) & "       "
StrMatrizEsqueleto(31, 1) = "sigblotc#" & Mid(StrModulo, 3, 2) & ":       "


For i = 1 To 33
    For j = 1 To 8
       StrLineaEns = StrLineaEns & StrMatrizEsqueleto(i, j)
    Next j
    StrLineaEns = StrLineaEns & vbCrLf
  Next i

  StrEnsResultadoMatriz = StrLineaEns


End Sub
```

Figura 4.2.11 Código en Visual Basic ® módulo lógico temporizador astable. (TEMPOE).

4.2.9 Esqueleto de programación y matriz de cadenas asociadas con el módulo lógico que realiza flip-flops asíncronos R-S (FFARS).

En la sección 3.2.9 del capítulo anterior se mostró el programa en lenguaje ensamblador asociado con el módulo lógico que realiza un flip-flop R-S asíncrono.

El programa ensamblador genérico se transcribe a continuación:

```
                lda tesppspt
                bne npptc#0
                bset bitQ,grupobitQ
                bra salidatc#0
npptc#0:        brset bitS,grupobitS,pon1sal#0
                brset bitR,grupobitR,pon0sal#0
                bra salidatc#0
pon1sal#0:      bset bitsal,grupobitsal
                bra salidatc#0
pon0sal#0:      bset bitsal,grupobitsal
                bra salidatc#0
salidatc#0:     nop
```

El esqueleto de programación puede ser construido con un arreglo matricial, donde los distintos elementos (i, j) de la matriz corresponden en posición y orden a los caracteres propios del programa genérico que resuelve cada módulo lógico realizable.

	1	2	3	4	5	6	7	8
1		lda		tesppspt				
2		bne		npptc#**nm**				
3		bset		**bitq**	,	**grupobitq**		
4		bra		salidatc#**nm**				
5	npptc#**nm**:	brset		**bits**	**,**	**grupobits**	,	pon1sal#**nm**
6		brset		**bitr**	**,**	**grupobitr**	,	pon0sal#**nm**
7		bra		salidatc#**nm**				
8	pon1sal#**nm**:	bset		**bitsal**	,	**grupobitsal**		
9		bra		**salidatc#nm**				
10	pon0sal#**nm**:	bset		**bitsal**	,	**grupobitsal**		
11		bra		salidatc#**nm**				
12	salidatc#**nm**:	nop						

Tabla 4.9 Matriz de cadenas asociada con el módulo que realizan flip-flops R-S asíncronos (FFARS).

Después el esqueleto de programación del módulo lógico realizable por él PLM08 será llamado por el programa GEN_ENS_PLM08 para hacer una sustitución de valores acorde entre los datos ingresados por el usuario al programar en lenguaje SIIL1 y las cadenas mudas vistas en la sección 3.2.9 del capítulo anterior. Generando con ello el programa en lenguaje ensamblador buscado para este caso.

Se hace una sustitución en la matriz de cadenas asociadas, denominada para el caso del código fuente del programa GEN_ENS_PLM08 como StrMatrizEsqueleto. Y una cadena denominada StrModulo. La cadena asociada StrModulo se genera al validar un archivo SIIL1 con el programa GEN_ENS_PLM08; esto para cada módulo lógico declarado por el usuario, (ver sección 4.3 de este capítulo, guía rápida del usuario).

La cadena StrModulo contiene información básica de las cadenas mudas a sustituir, para generar el archivo en código ensamblador para cada módulo lógico.

Por ejemplo; para un flip-flop R-S, cuyo bit de entrada S es el bit 7 del grupo 0 de entradas, su bit de reset R es el bit 6 del grupo 0 de entradas y su bit de salida Q es el bit 0 del grupo 0 de salidas. El nivel de verificación de las entradas R y S son bajos; se tiene prioridad para la variable booleana de entrada RESET y la salida se inicializa en nivel 0 lógico.

La sentencia SIIL1 correspondiente es la siguiente:

FFARS#22 E007,E006,S000,0000;

Para el caso de este módulo lógico, la cadena StrModulo es:

2122E007E006S0000000

Para la cual, el primer par de dígitos representa un código numérico, que indica se trata de un módulo lógico que realiza un flip-flop R-S asíncrono; el siguiente par de dígitos representa el número que el usuario asigno a este temporizador; los siguientes tres caracteres indican que la entrada de set “S” del temporizador pertenece al grupo cero de entradas, el siguiente carácter indica que el bit asociado con la entrada “S” es el bit 7; los siguientes tres caracteres indican que la entrada de reset “R” del temporizador pertenece al grupo cero de entradas, el siguiente carácter indica que el bit asociado con la entrada “R” es el bit 6. Los siguientes tres caracteres indican que la salida “Q” pertenece al grupo cero de salidas y el siguiente carácter indica que el bit físico de salidas, es el bit 0 del grupo antes mencionado. Los últimos cuatro dígitos indican que el nivel de verificación de las entradas R y S del temporizador están en nivel bajo, se tiene prioridad en la entrada de RESET y la salida se inicializa en cero lógico, respectivamente.

A continuación se muestra en la figura 4.2.12; el código en Visual Basic® que hace la asignación de cadenas mudas en la matriz correspondiente para el caso de un módulo lógico temporizador que realiza un flip-flop asíncrono R-S. (FFARS).

```
Public Sub FFARS(StrModulo As String)

Dim StrArchivo As String
Dim n As Integer
Dim StrLinea As String
Dim i As Long
Dim j As Long
Dim IntRenglones As Variant
Dim IntColumnas As Variant
Dim StrLineaEns As String
Dim Residuo As Integer
Dim HexaStr As String
Dim D As Long

ReDim StrMatrizEsqueleto(16, 21)

StrArchivo = App.Path & "\Templates\ffars.esq"
n = FreeFile()
Open StrArchivo For Input As #n
  Line Input #n, IntRenglones
  For i = 1 To IntRenglones
     Line Input #n, IntColumnas
     For j = 1 To IntColumnas
        Line Input #n, StrLinea
        StrMatrizEsqueleto(i, j) = StrLinea
     Next j
  Next i
Close #n


If (Mid(StrModulo, 17, 4)) = 0 Then       '0000
   StrMatrizEsqueleto(3, 2) = "bclr"
   StrMatrizEsqueleto(6, 2) = "brclr"
   StrMatrizEsqueleto(5, 2) = "brclr"
   StrMatrizEsqueleto(8, 2) = "bclr"
   StrMatrizEsqueleto(10, 2) = "bclr"

   StrMatrizEsqueleto(6, 4) = LCase(Mid(StrModulo, 8, 1))
   StrMatrizEsqueleto(6, 6) = "gp" & LCase(Mid(StrModulo, 5, 3))
   StrMatrizEsqueleto(5, 4) = LCase(Mid(StrModulo, 12, 1))
   StrMatrizEsqueleto(5, 6) = "gp" & LCase(Mid(StrModulo, 9, 3))

   End If
If (Mid(StrModulo, 17, 4)) = 1 Then       '0001 CKC
   StrMatrizEsqueleto(3, 2) = "bset"
   StrMatrizEsqueleto(6, 2) = "brclr"
   StrMatrizEsqueleto(5, 2) = "brclr"
   StrMatrizEsqueleto(8, 2) = "bclr"
   StrMatrizEsqueleto(10, 2) = "bset"

   StrMatrizEsqueleto(6, 4) = LCase(Mid(StrModulo, 8, 1))
   StrMatrizEsqueleto(6, 6) = "gp" & LCase(Mid(StrModulo, 5, 3))
   StrMatrizEsqueleto(5, 4) = LCase(Mid(StrModulo, 12, 1))
   StrMatrizEsqueleto(5, 6) = "gp" & LCase(Mid(StrModulo, 9, 3))
   End If


    .
    .
    .
  If (Mid(StrModulo, 17, 4)) = 1111 Then      '1111  NO CAMBIA ESQUELETO
    StrMatrizEsqueleto(3, 2) = "bset"        'CKC
     StrMatrizEsqueleto(5, 2) = "brset"
     StrMatrizEsqueleto(6, 2) = "brset"
     StrMatrizEsqueleto(8, 2) = "bset"
     StrMatrizEsqueleto(10, 2) = "bset"

     StrMatrizEsqueleto(5, 4) = LCase(Mid(StrModulo, 8, 1))
     StrMatrizEsqueleto(5, 6) = "gp" & LCase(Mid(StrModulo, 5, 3))
     StrMatrizEsqueleto(6, 4) = LCase(Mid(StrModulo, 12, 1))
     StrMatrizEsqueleto(6, 6) = "gp" & LCase(Mid(StrModulo, 9, 3))
     End If


  StrMatrizEsqueleto(3, 4) = LCase(Mid(StrModulo, 16, 1))
  StrMatrizEsqueleto(3, 6) = "gp" & LCase(Mid(StrModulo, 13, 3))
  StrMatrizEsqueleto(8, 4) = LCase(Mid(StrModulo, 16, 1))
  StrMatrizEsqueleto(8, 6) = "gp" & LCase(Mid(StrModulo, 13, 3))
  StrMatrizEsqueleto(10, 4) = LCase(Mid(StrModulo, 16, 1))
  StrMatrizEsqueleto(10, 6) = "gp" & LCase(Mid(StrModulo, 13, 3))

  StrMatrizEsqueleto(2, 4) = "npptc#" & Mid(StrModulo, 3, 2) & "        "
  StrMatrizEsqueleto(4, 4) = "salidatc#" & Mid(StrModulo, 3, 2) & "      "
  StrMatrizEsqueleto(5, 1) = "npptc#" & Mid(StrModulo, 3, 2) & ":       "
  StrMatrizEsqueleto(5, 8) = "pon1sal#" & Mid(StrModulo, 3, 2) & "       "
  StrMatrizEsqueleto(6, 8) = "pon0sal#" & Mid(StrModulo, 3, 2) & "       "
  StrMatrizEsqueleto(7, 4) = "salidatc#" & Mid(StrModulo, 3, 2) & "      "
  StrMatrizEsqueleto(8, 1) = "pon1sal#" & Mid(StrModulo, 3, 2) & ":       "
  StrMatrizEsqueleto(9, 4) = "salidatc#" & Mid(StrModulo, 3, 2) & "      "
  StrMatrizEsqueleto(10, 1) = "pon0sal#" & Mid(StrModulo, 3, 2) & ":       "
  StrMatrizEsqueleto(11, 4) = "salidatc#" & Mid(StrModulo, 3, 2) & "      "
  StrMatrizEsqueleto(12, 1) = "salidatc#" & Mid(StrModulo, 3, 2) & ":       "


  For i = 1 To 16
       For j = 1 To 8
          StrLineaEns = StrLineaEns & StrMatrizEsqueleto(i, j)
       Next j
       StrLineaEns = StrLineaEns & vbCrLf
    Next i

    StrEnsResultadoMatriz = StrLineaEns

End Sub
```

Figura 4.2.12 Código en Visual Basic ® módulo lógico que realizan flip-flops asíncronos R-S(FFARS).

4.2.10 Esqueleto de programación e implementación del módulo lógico que realiza contadores de eventos.

En la sección 3.2.10 del capítulo anterior se mostró el programa en lenguaje ensamblador asociado con el módulo lógico que realiza un contador de eventos.

El programa ensamblador genérico se transcribe a continuación:

```
                lda tesppspt
                bne npptc#0
resettc#0:      bclr bitsal,grupobitsal
inicontc#0:     ldhx #$FFFF
                sthx dirbconts+nn
                ldhx #$FFFF
                sthx dirbconts+nnn
                bra salidatc#0
npptc#0:        brset bitent,grupobitent,salidatc#0
                brclr bitrst,grupobitrst,resettc#0
                brset bitsal,grupobitsal,salidatc#0
                lda grupoedis+1
                and #$40
                sta locaux#1
                lda grupoedis
                and #$40
                cmp locaux#1
                beq salidatc#0
                bhi siguetc3#0
                bra salidatc#0
siguetc3#0:     ldhx dirbconts+nn
                aix #$FF
                sthx dirbconts+nn
                ldhx dirbconts+nn
                cphx dirbconts+nnn
                beq versaltc#0
salidatc#0:     bra sigblotc#0
versaltc#0:     bset bitsal,grupobitsal
                bra inicontc#0
sigblotc#0:     nop
```

El esqueleto de programación puede ser construido con un arreglo matricial, donde los distintos elementos (i, j) de la matriz corresponden en posición y orden a los caracteres propios del programa genérico que resuelve cada módulo lógico realizable.

	1	2	3	4	5	6	7	8
1		lda		tesppspt				
2		bne		npptc**#nm**				
3	resettc**#nm**:	bclr		**bitsal**	,	**grupobitsal**		
4	inicontc**#nm**:	ldhx		#$**FFFF**				
5		sthx		dirbconts+**nn**				
6		ldhx		#$**AAAA**				
7		sthx		dirbconts+**nnn**				
8		bra		salidatc**#nm**				
9	npptc**#nm**:	brset		**bitent**	,	**grupobitent**	,	salidatc**#nm**
10		brclr		**bitrst**	,	**grupobitrst**	,	resettc**#nm**
11		brset		**bitsal**	,	**grupobitsal**	,	salidatc**#nm**
12		lda		**grupoedis**+1				
13		and		#$40				
14		sta		locaux#1				
15		lda		**grupoedis**				
16		and		#$40				
17		cmp		locaux#1				
18		beq		salidatc**#nm**				
19		bhi		siguetc3**#nm**				
20		bra		salidatc**#nm**				
21	siguetc3**#nm**:	ldhx		dirbconts+**nn**				
22		aix		#$FF				
23		sthx		dirbconts+**nn**				
24		ldhx		dirbconts+**nn**				
25		cphx		dirbconts+**nnn**				
26		beq		versaltc**#nm**				
27	salidatc**#nm**:	bra		sigblotc**#nm**				
28	versaltc**#nm**:	bset		**bitsal**	,	**grupobitsal**		
29		bra		inicontc**#nm**				
30	sigblotc**#nm**:	nop						

Tabla 4.10 Matriz de cadenas asociada con el módulo contador de eventos.

Después el esqueleto de programación del módulo lógico realizable por él PLM08 será llamado por el programa GEN_ENS_PLM08 para hacer una sustitución de valores acorde entre los datos ingresados por el usuario al programar en lenguaje SIIL1 y las cadenas mudas vistas en la sección 3.2.10 del capítulo anterior. Generando con ello el programa en lenguaje ensamblador buscado para este caso.

Se hace una sustitución en la matriz de cadenas asociadas, denominada para el caso del código fuente del programa GEN_ENS_PLM08 como StrMatrizEsqueleto. Y una cadena denominada StrModulo. La cadena asociada StrModulo se genera al validar un archivo SIIL1 con el programa GEN_ENS_PLM08; esto para cada módulo lógico declarado por el usuario, (ver sección 4.3 de este capítulo, guía rápida del usuario).

La cadena StrModulo contiene información básica de las cadenas mudas a sustituir, para generar el archivo en código ensamblador para cada módulo lógico.

Por ejemplo; para la siguiente sentencia en SIIL1 que representa un contador de eventos, cuyo bit de entrada D es el bit 6 del grupo 1 de entradas; su entrada de congelamiento C es el bit 5 del grupo 1 de entradas; su bit de reset es el bit 4 del grupo 1 de entradas y su salida es el bit 7 del grupo 0 de salidas.

Sus características: se modifica la cuenta descendente en la entrada de disparo D; el nivel de verificación de la entrada C es bajo; el nivel de verificación de la entrada R es alto y el nivel de verificación de su salida es alto. Su cuenta comienza en 10 y termina en 5.

La sentencia SIIL1 correspondiente es la siguiente:

CONTA#14 E016,E015,E014,S007,00010,00005,00001;

Para el caso de este módulo lógico, la cadena StrModulo es:

2214E016E015E014S007000100000500001

Para la cual, el primer par de dígitos representa un código numérico, que indica se trata de un módulo lógico contador de eventos; el siguiente par de dígitos representa el número que el usuario asigno a este temporizador; los siguientes tres caracteres indican que la entrada de disparo “D” del temporizador pertenece al grupo uno de entradas, el siguiente carácter indica que el bit asociado con la entrada “D” es el bit 6. Los siguientes tres caracteres indican que la entrada de congelamiento “C” del temporizador pertenece al grupo uno de entradas, el siguiente carácter indica que el bit asociado con la entrada “C” es el bit 5. Los siguientes tres caracteres indican que la entrada de reset “R” del temporizador pertenece al grupo uno de entradas, el siguiente carácter indica que el bit asociado con la entrada “R” es el bit 4. Los siguientes tres caracteres indican que la salida “T” pertenece al grupo cero de salidas y el siguiente carácter indica que el bit físico de salidas, es el bit 7 del grupo antes mencionado. Los siguientes cinco caracteres indican la cifra de la cuenta inicial es 10; los siguientes cinco caracteres indican la cifra de la cuenta final igual a cinco. Los últimos 5 dígitos significan que se modifica la cuenta para flancos de bajada, el nivel de verificación de la entrada C es bajo, el nivel de verificación de la entrada R es alto, el contador es descendente y el nivel de verificación de la salida TF es bajo, respectivamente.

A continuación se muestra en la figura 4.2.13; el código en Visual Basic® que hace la asignación de cadenas mudas en la matriz correspondiente para el caso de un módulo lógico contador de eventos.

```
Public Sub Contador(StrModulo As String)

Dim StrArchivo As String
Dim n As Integer
Dim StrLinea As String
Dim i As Long
Dim j As Long
Dim IntRenglones As Variant
Dim IntColumnas As Variant
Dim StrLineaEns As String
Dim Residuo As Integer
Dim HexaStr As String
Dim D As Long

ReDim StrMatrizEsqueleto(32, 21)

StrArchivo = App.Path & "\Templates\contador.esq"
n = FreeFile()
Open StrArchivo For Input As #n
  Line Input #n, IntRenglones
  For i = 1 To IntRenglones
    Line Input #n, IntColumnas
    For j = 1 To IntColumnas
      Line Input #n, StrLinea
      StrMatrizEsqueleto(i, j) = StrLinea
    Next j
  Next i
Close #n
                    'RUTINA RELOJ CUENTA INICIAL

                    'RUTINA RELOJ 2

If (Mid(StrModulo, 31, 5)) = 0 Then     '00000  CADENA VALIDA 2214E007E006E005S000000000000701010 CKC
  StrMatrizEsqueleto(3, 2) = "bset"
  StrMatrizEsqueleto(9, 2) = "brclr"
  StrMatrizEsqueleto(10, 2) = "brset"
  StrMatrizEsqueleto(11, 2) = "brclr"
  StrMatrizEsqueleto(19, 2) = "blo"
  StrMatrizEsqueleto(28, 2) = "bclr"
'  StrMatrizEsqueleto(22, 4) = "#$FF"    'D=0
  End If

If (Mid(StrModulo, 31, 5)) = 1 Then     '00001 CKC
  StrMatrizEsqueleto(3, 2) = "bclr"
  StrMatrizEsqueleto(9, 2) = "brclr"
  StrMatrizEsqueleto(10, 2) = "brset"
  StrMatrizEsqueleto(11, 2) = "brset"
  StrMatrizEsqueleto(19, 2) = "blo"
  StrMatrizEsqueleto(28, 2) = "bset"
'  StrMatrizEsqueleto(22, 4) = "#$FF"    'D=0
  End If

    .
    .
    .
  If (Mid(StrModulo, 31, 5)) = 11110 Then     '11110 CKC
    StrMatrizEsqueleto(3, 2) = "bset"
    StrMatrizEsqueleto(9, 2) = "brset"
    StrMatrizEsqueleto(10, 2) = "brclr"
    StrMatrizEsqueleto(11, 2) = "brclr"
    StrMatrizEsqueleto(19, 2) = "bhi"
    StrMatrizEsqueleto(28, 2) = "bclr"
    StrMatrizEsqueleto(22, 4) = "#$01"   'D=1

    End If
  If (Mid(StrModulo, 31, 5)) = 11111 Then     '11111 CKC
    StrMatrizEsqueleto(3, 2) = "bclr"
    StrMatrizEsqueleto(9, 2) = "brset"
    StrMatrizEsqueleto(10, 2) = "brclr"
    StrMatrizEsqueleto(11, 2) = "brset"
    StrMatrizEsqueleto(19, 2) = "bhi"
    StrMatrizEsqueleto(28, 2) = "bset"
    StrMatrizEsqueleto(22, 4) = "#$01"   'D=1

    End If
```

```
StrMatrizEsqueleto(3, 4) = LCase(Mid(StrModulo, 20, 1))
StrMatrizEsqueleto(3, 6) = "gp" & LCase(Mid(StrModulo, 17, 3))

StrMatrizEsqueleto(4, 4) = "#$" & E8

StrMatrizEsqueleto(5, 4) = "dirbconts+" & nn
StrMatrizEsqueleto(6, 4) = "#$" & C8

StrMatrizEsqueleto(7, 4) = "dirbconts+" & nnn

StrMatrizEsqueleto(9, 4) = LCase(Mid(StrModulo, 12, 1))
StrMatrizEsqueleto(9, 6) = "gp" & LCase(Mid(StrModulo, 9, 3))

StrMatrizEsqueleto(10, 4) = LCase(Mid(StrModulo, 16, 1))
StrMatrizEsqueleto(10, 6) = "gp" & LCase(Mid(StrModulo, 13, 3))
StrMatrizEsqueleto(11, 4) = LCase(Mid(StrModulo, 20, 1))
StrMatrizEsqueleto(11, 6) = "gp" & LCase(Mid(StrModulo, 17, 3))

StrMatrizEsqueleto(12, 4) = "gp" & LCase(Mid(StrModulo, 5, 3)) & "+1"
StrMatrizEsqueleto(13, 4) = "#$" & Y
StrMatrizEsqueleto(15, 4) = "gp" & LCase(Mid(StrModulo, 5, 3))
StrMatrizEsqueleto(16, 4) = "#$" & Y

StrMatrizEsqueleto(21, 4) = "dirbconts+" & nn
StrMatrizEsqueleto(23, 4) = "dirbconts+" & nn
StrMatrizEsqueleto(24, 4) = "dirbconts+" & nn
StrMatrizEsqueleto(25, 4) = "dirbconts+" & nnn

StrMatrizEsqueleto(28, 4) = LCase(Mid(StrModulo, 20, 1))
StrMatrizEsqueleto(28, 6) = "gp" & LCase(Mid(StrModulo, 17, 3))

StrMatrizEsqueleto(2, 4) = "npptc#" & Mid(StrModulo, 3, 2) & "         "
StrMatrizEsqueleto(3, 1) = "resettc#" & Mid(StrModulo, 3, 2) & ":       "
StrMatrizEsqueleto(4, 1) = "inicontc#" & Mid(StrModulo, 3, 2) & ":      "
StrMatrizEsqueleto(8, 4) = "salidatc#" & Mid(StrModulo, 3, 2) & "       "
StrMatrizEsqueleto(9, 1) = "npptc#" & Mid(StrModulo, 3, 2) & ":         "
StrMatrizEsqueleto(9, 8) = "salidatc#" & Mid(StrModulo, 3, 2) & "       "
StrMatrizEsqueleto(10, 8) = "resettc#" & Mid(StrModulo, 3, 2) & "       "
StrMatrizEsqueleto(11, 8) = "salidatc#" & Mid(StrModulo, 3, 2) & "      "

StrMatrizEsqueleto(18, 4) = "salidatc#" & Mid(StrModulo, 3, 2) & "      "
StrMatrizEsqueleto(19, 4) = "siguetc3#" & Mid(StrModulo, 3, 2) & "      "
StrMatrizEsqueleto(20, 4) = "salidatc#" & Mid(StrModulo, 3, 2) & "      "
StrMatrizEsqueleto(21, 1) = "siguetc3#" & Mid(StrModulo, 3, 2) & ":     "
StrMatrizEsqueleto(26, 4) = "versaltc#" & Mid(StrModulo, 3, 2) & "      "
StrMatrizEsqueleto(27, 1) = "salidatc#" & Mid(StrModulo, 3, 2) & ":      "
StrMatrizEsqueleto(27, 4) = "sigblotc#" & Mid(StrModulo, 3, 2) & "      "
StrMatrizEsqueleto(28, 1) = "versaltc#" & Mid(StrModulo, 3, 2) & ":      "
StrMatrizEsqueleto(29, 4) = "inicontc#" & Mid(StrModulo, 3, 2) & "      "
StrMatrizEsqueleto(30, 1) = "sigblotc#" & Mid(StrModulo, 3, 2) & ":      "

For i = 1 To 32
    For j = 1 To 8
      StrLineaEns = StrLineaEns & StrMatrizEsqueleto(i, j)
    Next j
    StrLineaEns = StrLineaEns & vbCrLf
  Next i

  StrEnsResultadoMatriz = StrLineaEns

End Sub
```

Figura 4.2.13 Código en Visual Basic ® módulo lógico que realiza un contador de eventos.

4.3 Guía rápida del usuario del programa generador de lenguaje ensamblador para el PLM08 (GEN_ENS_PLM08).

A continuación se presenta una guía breve, para que el usuario tenga las instrucciones básicas de cómo manejar el GEN_ENS_PLMM08 de manera apropiada.

4.3.1 Abrir un archivo.

Para abrir un archivo con la extensión SIL se utiliza el botón Abre archivo Sil. Con esta acción aparecerá inmediatamente una ventana que permite buscar el archivo que se desea abrir como se muestra en la figura 4.3 y en la figura 4.4 se muestra como se presenta en una pantalla la transcripción del archivo de interés.

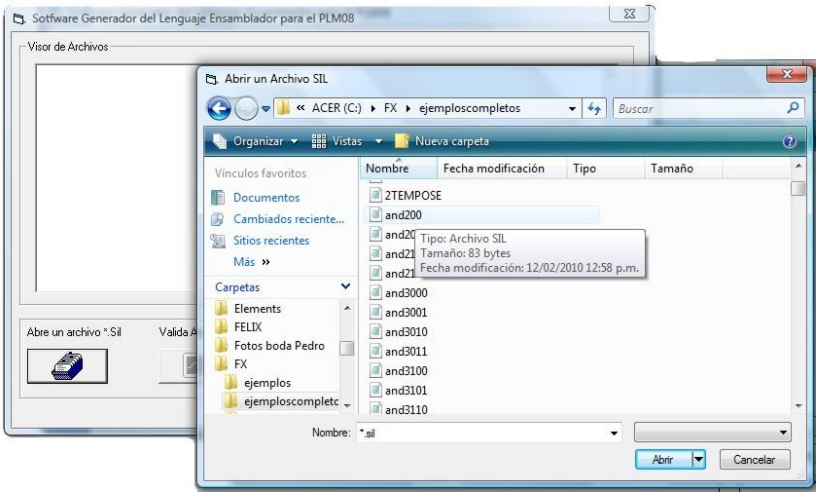


Figura 4.3 Botón abre archivo Sil.

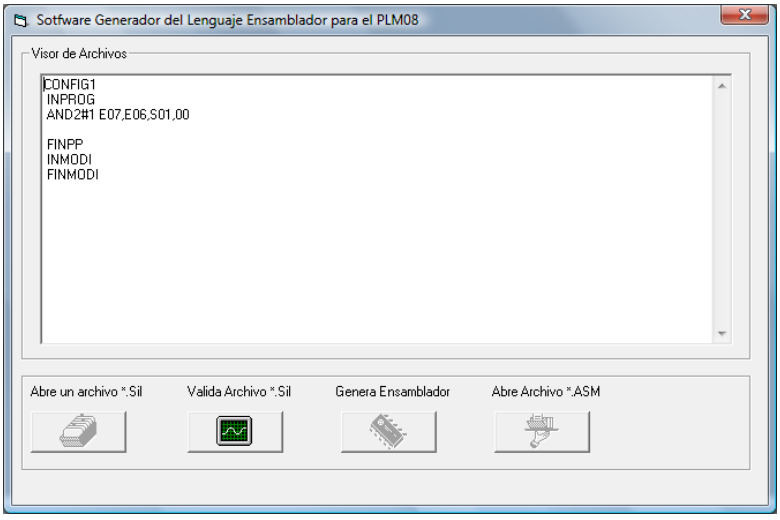


Figura 4.4 Ventana que se abre para visualizar el archivo de interés.

Una vez ejecutada la acción de búsqueda se activa en el panel el botón valida archivo. Sil (figura 4.5) que al ser pulsado activa el uso del compilador auxiliar vc3v.bas, desarrollado en lenguaje BASIC por el M.I Antonio Salvá Calleja. Este compilador se adaptó al proceso de análisis del GEN_ENS_PLM08 para detectar los mismos tipos de errores e inconsistencias de sintaxis en el código SIIL1 que se filtran para el PLM, y su ejecución se realiza a nivel de DOS (Sistema Operativo de Disco) en una línea de comando. Los resultados de compilación se obtienen por medio de dos archivos de texto que son inspeccionados por el GEN_ENS_PLM08. Con la información obtenida, el sistema determina la necesidad de hacer ajustes o modificaciones por parte del usuario; este proceso se explica a continuación:

Figura 4.5 botón valida archivo.

4.3.2 Reporte de errores.

En la figura 4.6 se muestra un archivo .SIL correspondiente a una compuerta AND2, en este programa fuente se escribieron errores de sintaxis para mostrar el funcionamiento del reporte de errores.

Durante el proceso de análisis de un programa fuente, el compilador auxiliar de código SIIL1 puede detectar errores en la declaración de uno o más módulos lógicos o en los limitadores de los subprogramas principal y temporizado. Si esta situación ocurre, el sistema presenta al usuario una pantalla alterna en la cual se indican los errores de sintaxis que se encontraron dentro del programa fuente; dicha pantalla se divide en dos partes que muestran información relacionada con los errores y contiene además en la parte inferior dos botones para revisar el código y guardar los cambios, como se muestra en la figura 4.7.

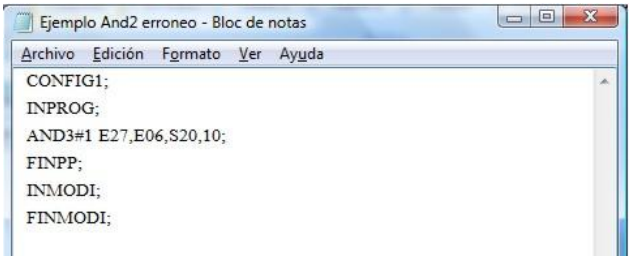


Figura 4.6 Programa fuente con errores de sintaxis.

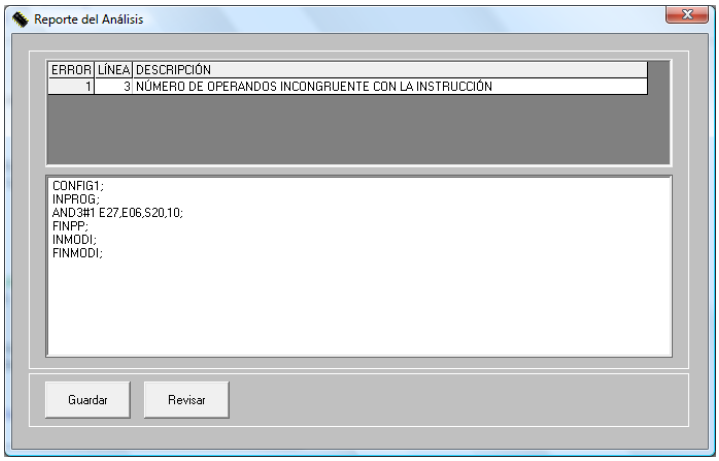


Figura 4.7 Pantalla de reporte de errores.

Como se observa, la parte superior de esta pantalla contiene una lista de errores encontrados, detallando el tipo, la línea del programa fuente en que se localiza y la descripción del error; la parte inferior del reporte de errores contiene el texto completo del programa fuente para ser editado por el usuario y corregir el error. Después que el usuario ha hecho cambios pertinentes en su código, y guardado dichos cambios presionando el botón guardar, puede analizar nuevamente el programa pulsando el botón revisar.

Una vez que se ha revisado el código fuente y el compilador no encontró más errores, esta pantalla desaparece para dar lugar a la presentación de un cuadro de dialogo que reporta al usuario el número de módulos detectados en los programas principal y temporizado de la aplicación como puede verse en la figura 4.8.

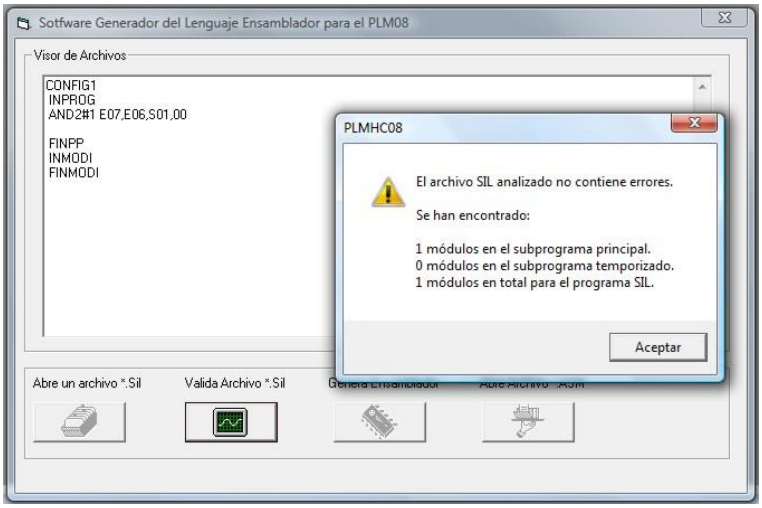


Figura 4.8. Reporte del análisis.

4.3.3 Genera programa ensamblador.

Una vez validado, se activa el botón genera ensamblador (figura 4.9), que hace que el programa GEN_ENS_PLM08 haga una sustitución de la información que requieren las cadenas mudas que se explicaron en las secciones 3.2.1 a 3.2.10 del capítulo anterior. Y mediante el uso de los esqueletos genéricos en lenguaje ensamblador para cada compuerta y el esqueleto de estructura general, den forma al programa final buscado por el usuario según sea el caso.

Figura 4.9 Botón genera ensamblador.

Una vez presionado el botón, se abre un cuadro de dialogo que testifica que el programa se ha generado (figura 4.10). Y con esta acción ha quedado grabado en la carpeta denominada con el nombre *salidas,* ubicada en la carpeta raíz donde está instalado el programa GEN_ENS_PLM08.

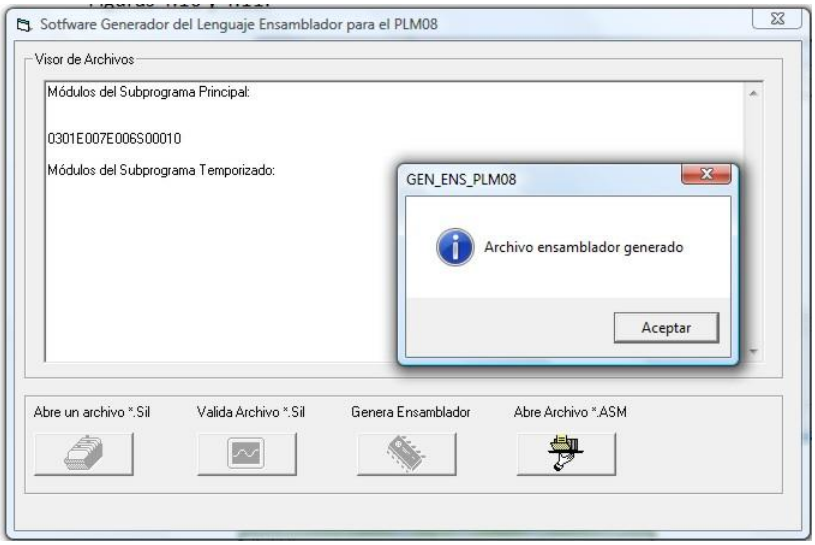


Figura 4.10 Dialogo que muestra el aviso de que el programa en lenguaje ensamblador se ha generado.

4.3.4 Abrir archivo .ASM

Este botón (véase figura 4.11), sirve para hacer visible al usuario la generación de su código en lenguaje ensamblador, al pulsarlo; la sustitución de toda la información programada por el usuario se despliega en un archivo de bloc de notas (figura 4.12), el cual posteriormente puede ser revisado por el usuario o cargado en el programa PUMMA_08+, desarrollado por el M.I Antonio Salvá Calleja y cuya información completa de uso se encuentra disponible en internet en la dirección:
http://dctrl.fi-b.unam.mx/~salva/MUAIDA08b_enero_2010.pdf para su posterior ejecución con la tarjeta de desarrollo MINICON_O8A.

Figura 4.11 Botón abre archivo .ASM

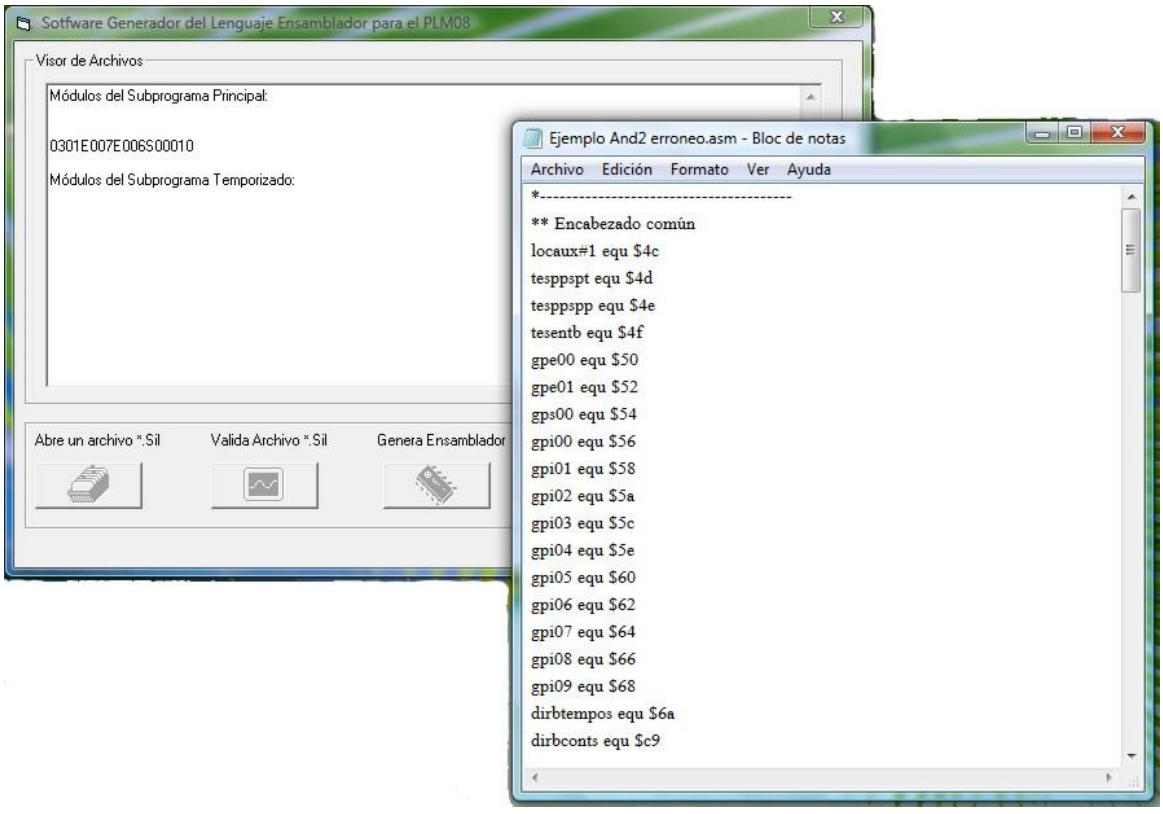


Figura 4.12 Programa completo generado por GEN_ENS PLM08.

Referencias:

- Altamirano Yépez Luis Antonio, Dehesa Castillejos Erick Abraham, Hernandez Reyes Maricarmen –"Desarrollo de software de simulación para el PLM (Programador lógico modular). Tesis de licenciatura, Facultad de Ingeniería, UNAM, 2003.
- Duncan Mackenzie – "Aprendiendo visual basic.net". Ed. Pearson educación, México, 2003.