

LABERINTO ASM-6809

*2019-2020
COMPUTADORES II*

Javier García Pechero

[Redacted]

Germán Francés Tostado

[Redacted]

LABERINTO_MAIN.ASM

Este fichero contiene el grueso del programa, comienza con la inicialización de todas las variables a cero y las cadenas de texto que usaremos posteriormente, desde menús hasta mensajes de error.

```
2 .module laberinto_main
3 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; definicion de constantes,cadenas,variables ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
4 fin .equ 0xFF01
5 teclado .equ 0xFF02
6 pantalla .equ 0xFF00
7 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; VARIABLES ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
8 temp: .word 0
9 lv: .byte 0
10 option: .byte 0
11 contadormov: .word 0
12 tecla: .byte 0
13 direccion_anterior: .byte 0
14 guardado: .byte 0
15 direccion_guardado: .word 0
16 direccion_imprimir: .word 0
17 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; MENSAJES Y MENUS ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
18 presentacion_mp: .ascii "\n-----+\n"
19 .ascii "| Laberinto | \n"
20 .ascii "+-----+\n"
21 .ascii "\33[37m1)\33[33m ELEGIR LABERINTO\n"
22 .ascii "\33[37m2)\33[33m JUGAR\n"
23 .ascii "\33[37m3)\33[33m SALIR\n"
24 .byte 0
25 elija_opcion: .asciz "\nIntroduzca una opcion:"
26
27 elegir_lab: .asciz "Elija un laberinto "
28
29 opciones: .asciz "(1-"
30
31 parentesis: .asciz ");:"
32
33 opcion_erronea: .asciz "\nOpcion erronea, introduzca de nuevo: "
34
35 msg_error: .asciz "\nOpcion erronea, introduzca una opcion valida\n"
36
37 elija_laberinto: .asciz "\n\nELEGIR LABERINTO\n"
38
39
40 lbtr_seleccionado: .asciz "\nHa elegido el laberinto numero "
41
42 msg_salida: .asciz "\n\nHa salido correctamente.\n\n\33[0m"
43
44
45 presentacion_menu_2: .ascii "\n1) Volver al menu principal\n"
46 .ascii "\n2) Jugar\n"
47 .ascii "Opcion: "
48 .byte 0
49
50 menu_next_lv: .ascii "\n1) Volver al menu principal\n"
51 .ascii "\n2) Jugar siguiente nivel\n"
52 .ascii "Opcion: "
53 .byte 0
54
55
```

Al ser un programa completo, decidimos dividirlo de manera que tuviéramos un segundo fichero con subrutinas que llamaríamos varias veces a lo largo de la ejecución. Estando estas subrutinas en un nuevo documento, se declaran de manera global.

```
4 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; PROGRAMA ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
5 .globl programa
6 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; BIBLIOTECA DE SUBROUTINAS ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
7 .globl imprime_cadena
8 .globl imprime_laberinto
9 .globl laberintos
10 .globl laberinto_numero
11 .globl laberinto_tamano
12 .globl imprime_laberinto_especifico
13 .globl calculo_pos_inicial
14 .globl calculo_desplazamiento
15 .globl imprime_numeros
16 .globl calculo_pos_final
17 .globl limpia_laberinto_especifico
18 .globl calculos
19 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; CONSTANTES COLORES Y OTROS EFECTOS TEXTO ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
20 .globl clearScreen ;Limpiar pantalla
21 .globl setNormal
22 .globl setNegative
23 .globl setUnderlined
24 .globl setYellow
25 .globl setBlue
26 .globl setRed
27 .globl setGreen
28 .globl setBold
29 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; VARIABLES ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
30 .globl option
31 .globl lv
32 .globl temp
33 .globl contadormov
34 .globl col
35 .globl fil
36 .globl pos_inicio
37 .globl direccion_inicio
38 .globl direccion_final
39 .globl base_tablero
40 .globl desplazamiento
41 .globl desplazamiento_final
42 .globl direccion_anterior
43 .globl direccion_guardado
```

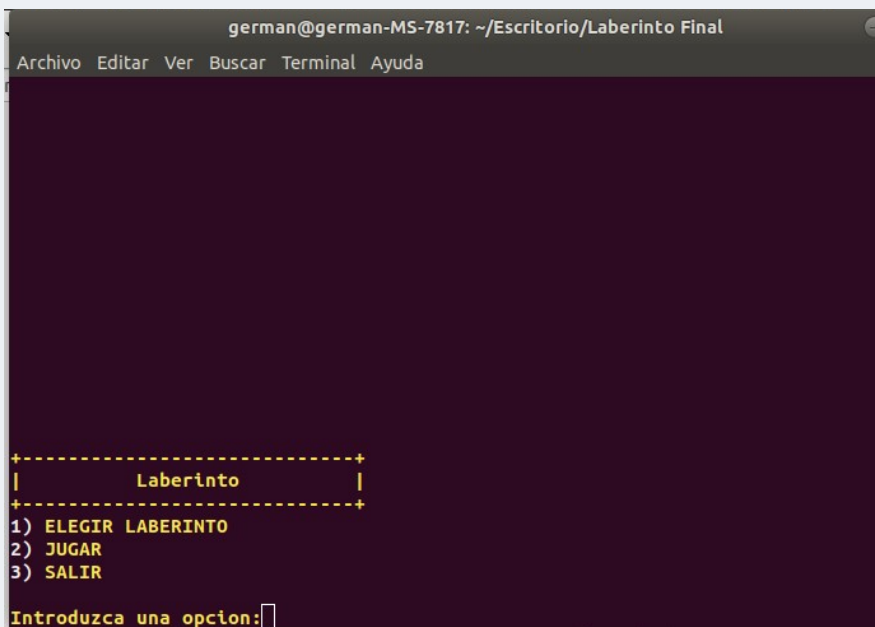
Comienza con la presentación del menú principal, a partir del cual se desarrollará el resto.

Este menú principal se muestra por pantalla y recoge por teclado la respuesta. Si es una válida, saltará a la opción escogida, sino, vuelve a pedir.

Aprovechamos esta sección de código para poner el segundo menú y así tener ambos en la misma porción de código.

El segundo menú se encargará de que el usuario ratifique si quiere jugar con el laberinto seleccionado o volver al menú principal.

```
140 ;;;;;;;;;;;;;;;;;;;;;;;;;; INICIO MAIN ;;;;;;;;;;;;;;;;;;;;;;;;;;
141 main:
142         ldd     #0
143         std     temp
144         ldb     #1
145         stb     lv
146         clra
147         clrb
148 menu_principal:
149         ldx     #clearScreen
150         jsr     imprime_cadena
151         ldx     #setBold
152         jsr     imprime_cadena
153         ldx     #setYellow
154         jsr     imprime_cadena
155         ldx     #presentacion_mp
156         jsr     imprime_cadena
157         ldx     #elija_opcion
158         jsr     imprime_cadena
159 mp_ask_sgte:
160         lda     teclado
161         cmpa    #'1
162         beq     opcion1
163         cmpa    #'2
164         lbeq    jugardirect
165         cmpa    #'3
166         lbeq    opcion3
167         ldx     #opcion_erronea
168         jsr     imprime_cadena
169         bra     mp_ask_sgte
170 menu_2:
171         ldx     #presentacion_menu_2
172         jsr     imprime_cadena
173 m2_ask_sgte:
174         lda     teclado
175         cmpa    #'1
176         beq     menu_principal
177         cmpa    #'2
178         lbeq    opcion2
179         ldx     #opcion_erronea
180         jsr     imprime_cadena
181         bra     m2_ask_sgte
```



Menú Principal

El resto del programa se divide en dos grandes partes:

- Opción 1, Elegir Laberinto
- Opción 2, Jugar

OPCIÓN 1, ELEGIR LABERINTO

Primero se limpian los laberintos mediante un bucle for, ya que si se desea reiniciar el juego una vez terminado hay que limpiar todo el recorrido hecho por el jugador anteriormente para empezar de cero.

La variable 'lv' mantiene el número de laberinto a limpiar, desde uno hasta n.

Salta a diferentes subrutinas que están en otro fichero que veremos más adelante.

Imprimirá todos los laberintos disponibles e indicará el número de cada uno.

De forma similar a antes, pedirá una respuesta dentro de las contempladas.

```

LABERINTO: 3
XXXXXXXXXXXXXXXXX X
X  X X  X  X
X X  X XXXX X X
X XXX X X X X X
X  X X X X XXXX
X X XXX      X
X X      XX XX XX
XXX XX XXX X XX
X  XXXX  X X
X X      XXXX XX
X XX X X X  X
X X  X X XXXXX
XX X X X  X X
XXXX XXXXX X X X
X      X X  X
XXXXXXXXXXXXXXXXX X
Elija un laberinto (1-3):
    
```

Ejemplo en la terminal

```

182 ;;;;;;;;;;;;;;;;;;;;;;;;;; OPCION 1 = ELEGIR LABERINTO
183 opcion1:
184         clrb
185         ldb    #1
186         stb    lv
187         ldd    #0
188         std    temp
189 b_for_limpiar_elegirlb:
190         ldb    lv
191         cmpb   laberinto_numero
192         bhi    elegirlb
193         jsr    calculos
194         ldx    #laberintos
195         jsr    limpia_laberinto_especifico
196         clrb
197         ldb    lv
198         incb
199         stb    lv
200         clrb
201         bra    b_for_limpiar_elegirlb
202 elegirlb:
203         ldx    #elija_laberinto
204         jsr    imprime_cadena
205         ldx    #setBlue
206         jsr    imprime_cadena
207         ldx    #laberintos
208         lda    #1
209 b_for_imprimir:
210         stx    direccion_imprimir
211         cmpa   laberinto_numero
212         bhi    ask_sgte
213         ldx    #setYellow
214         jsr    imprime_cadena
215         ldx    #nivel
216         jsr    imprime_cadena
217         ldx    #setRed
218         jsr    imprime_cadena
219         adda   #'0
220         sta    pantalla
221         suba   #'0
222         ldx    #setBlue
223         jsr    imprime_cadena
224         ldx    direccion_imprimir
225         jsr    imprime_laberinto
226         inca
227         bra    b_for_imprimir
228 ask_sgte:
229         ldx    #setYellow
230         jsr    imprime_cadena
231         ldx    #elegir_lab
232         jsr    imprime_cadena
233         ldx    #opciones
234         jsr    imprime_cadena
235         lda    laberinto_numero
236         adda   #'0
237         sta    pantalla
238         ldx    #parentesis
239         jsr    imprime_cadena
240 bwhile_opcion:
241         ldb    teclado
242         subb   #'0
243         cmpb   #0
244         ble    repetir_while
245         cmpb   laberinto_numero
246         ble    w_seguir
247 repetir_while:
248         ldx    #opcion_erronea
249         jsr    imprime_cadena
250         bra    bwhile_opcion
251 w_seguir:
252         lda    #0
253         sta    guardado
254         sta    contadormov
255         ldx    #clearScreen
256         jsr    imprime_cadena
257         ldx    #lbrt_seleccionado
258         jsr    imprime_cadena
259         addb   #'0
260         ldx    #setRed
261         jsr    imprime_cadena
262         stb    pantalla
263         subb   #'0
264         stb    lv
265         ldx    #setYellow
266         jsr    imprime_cadena
267         bra    lbrt
    
```

Al final de la opción uno, hace los cálculos pertinentes al laberinto seleccionado, y lo muestra por pantalla. Estos cálculos sacan la posición inicial del tablero a partir de la cual imprimir, por si se selecciona algún laberinto diferente al primero.

```

269 jugardirect:
270         ldx    #clearScreen
271         jsr    imprime_cadena
272 lbrt:
273         clra
274         jsr    calculos
275 next:
276         ldx    #setYellow
277         jsr    imprime_cadena
278         ldx    #nivel
279         jsr    imprime_cadena
280         ldx    #setRed
281         jsr    imprime_cadena
282         ldb    lv
283         addb   #'0
284         stb    pantalla
285         clrb
286         ldb    #'\\n
287         stb    pantalla
288         clrb
289         ldx    #setBlue
290         jsr    imprime_cadena
291         ldx    #laberintos
292         jsr    imprime_laberinto_especifico
293         ldx    #setYellow
294         jsr    imprime_cadena
295         lbra   menu_2

```

Ejemplo en la terminal tras seleccionar un laberinto

```

german@german-MS-7817: ~/Es
Archivo Editar Ver Buscar Terminal Ayuda

Ha elegido el laberinto numero 2
LABERINTO: 2
X XXXXXXXXXXXXXXXX
X  X X  X  X
X X  X XXXX X X
X XXX X X X X X
X  X  X X XX X
X X XXX X  X
X X  XX XX XX
XXX XX XXX X XX
X  XX X  X X
X X  XXXX XX
X XX X X X  X
X  X  X X XXXX
XX X X X  X X
XXXX XXXXX X X X
X      X X X
XX XXXXXXXXXXXXXXXX

1) Volver al menu principal
2) Jugar
Opcion: 

```


OPCIÓN 2, JUGAR

Comienza con la comprobación de que no haya una partida anterior guardada mediante un flag.

Si guardado = 1 significa que hay que reanudar un juego anterior.

Todo este proceso con las variables concretas está detallado dentro del código mediante comentarios.

```
296 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; OPCION 2 = JUGAR;;;;;;;;;;;;;;;;;
297 opcion2:
298         clra
299         lda     guardado
300         cmpa    #1
301         beq     continuar_guardado
302         jsr     calculo_pos_inicial
303         jsr     calculo_pos_final
304         ldy     direccion_inicio
305         sty     direccion_anterior
306         sty     direccion_guardado
307         sty     desplazamiento_final
308         bra     jugar
309 continuar_guardado:
310         ldy     direccion_guardado
311         sty     direccion_anterior
312         bra     jugar
```

```
367         lbeq    next_lb
368 pedir_movimiento:
369         ldx     #setYellow
370         jsr     imprime_cadena
371         ldx     #teclas
372         jsr     imprime_cadena
373         clra
374         lda     teclado
375         sta     tecla
376 comprobar_mov:
377         cmpa    #'q
378         beq     arriba
379         cmpa    #'p
380         beq     derecha
381         cmpa    #'o
382         beq     izquierda
383         cmpa    #'a
384         beq     abajo
385         cmpa    #'x
386         lbeq    querer_ir_menu
387         ldx     #opcion_erronea
388         jsr     imprime_cadena
389         lbra     error
```

Tras imprimir el laberinto adecuado y haber colocado al jugador en la posición inicial, va pidiendo los movimientos al jugador. En base a las respuestas comprueba si es válido (está dentro de los controles) y si lo es, comprueba si ese mismo movimiento puede hacerse.

Dependiendo de la dirección escogida se incrementará o decrementará una fila o columna. Y en caso de que se encuentre una pared, se corregirá esta última operación.

Asimismo, se van imprimiendo puntos en las direcciones anteriores del jugador, guardadas en la variable 'direccion_anterior', si encuentra un punto, sumará 5 al contador de movimientos.

Este bucle de jugar y pedir movimientos persistirá hasta que se encuentre en el espacio en blanco de la primera fila, almacenado en 'direccion_final' tras localizarlo en una subrutina.

```
german@german-MS-7817: ~/Escritorio/Laberinto Final
Archivo Editar Ver Buscar Terminal Ayuda

LABERINTO: 3
Movimientos:030
XXXXXXXXXXXXXXXXX X
X  X X  X  X
X X  X XXXX X X
X XXX X X X  X X
X  X X X X XXXX
X X XXX  o  X
X X  XX.XX XX
XXX XX XXX..X XX
X  XXXX...X X
X X  .XXXX XX
X XX X X.X  X
X X  X.X XXXXX
XX X X X.....X X
XXXX XXXXX X.X X
X  X X...X
XXXXXXXXXXXXXXXXX.X

(o\p\q\A)(X):
```

Ejemplo del juego en ejecución.

Una vez terminados los niveles pedirá si quiere pasar al siguiente nivel o volver al menú principal.

Si se completa el último nivel mostrará el mensaje de 'Juego completado' y el usuario decidirá si reiniciar los tableros o salir del programa.

SUBROUTINAS.ASM

Este fichero contiene las subrutinas que se emplean en el main.

```
68 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
69 ; imprime_cadena ;
70 ; saca por la pantalla la cadena acabada en '\0 apuntada por X ;
71 ; ;
72 ; Entrada: X-direccion de comienzo de la cadena ;
73 ; Salida: ninguna ;
74 ; Registros afectados: X, CC. ;
75 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
76 imprime_cadena:
77     pshu    a,x
78 ic_sgte:
79     lda     ,x+
80     beq     ret_imprime_cadena
81     sta     pantalla
82     bra     ic_sgte
83 ret_imprime_cadena:
84     pulu    a,x
85     rts
86 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
87 ; imprime_laberinto ;
88 ; saca por la pantalla un laberinto y añade \n cada linea ;
89 ; ;
90 ; Entrada: X-direccion de comienzo de la cadena ;
91 ; Salida: ninguna ;
92 ; Registros afectados: CC. ;
93 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
94 imprime_laberinto:
95     pshu    a
96     ldb     #' \n
97     stb     pantalla
98     clrb
99 il_sgte:
100    lda     ,x+
101    beq     rts_imprime_laberinto
102    sta     pantalla
103    incb
104    cmpb    laberinto_tamano
105    beq     il_retorno
106
107    bra     il_sgte
108 il_retorno:
109    clrb
110    ldb     #' \n
111    stb     pantalla
112    clrb
113    bra     il_sgte
114 rts_imprime_laberinto:
115    pulu    a
116    rts
117 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

imprime_laberinto:

Se basa en el mismo principio que imprime_cadena pero al final de cada fila (laberinto_tamano) añade un retorno de carro.


```

117 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
118 ; imprime_laberinto_especifico ;
119 ; saca por la pantalla un laberinto concreto ;
120 ; y añade \n cada linea ;
121 ; ;
122 ; Entrada: X-direccion de comienzo de la cadena ;
123 ; Salida: ninguna ;
124 ; Registros afectados: CC. ;
125 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
126 imprime_laberinto_especifico:
127     ldd     temp
128     leax    d,x
129     clra
130     pshu    a,x
131     clrb
132     stx     direccion_auxiliar
133 ile_sgte:
134     lda     ,x+
135     beq     rts_imprime_laberinto_especifico
136     sta     aux
137     stx     direccion_auxiliar
138     cmpa    #'o
139     beq     cambiaverde
140     cmpa    #'.'
141     beq     cambiaverde
142     bra     ile_sigue
143 cambiaverde:
144     ldx     #setGreen
145     jsr     imprime_cadena
146 ile_sigue:
147     lda     aux
148     sta     pantalla
149     ldx     #setBlue
150     jsr     imprime_cadena
151     ldx     direccion_auxiliar
152     incb
153     cmpb    laberinto_tamano
154     beq     ile_retorno
155     bra     ile_sgte
156 ile_retorno:
157     clrb
158     ldb     #' \n
159     stb     pantalla
160     clrb
161     bra     ile_sgte
162 rts_imprime_laberinto_especifico:
163     pulu    a,x
164     rts

```

imprime_laberinto_especifico:

*Una variación más útil de
imprime_laberinto, que es la que usamos durante el juego.*

*Imprime el laberinto concreto con el que se está jugando, ya que
hemos ido adoptando diferentes colores para la claridad del juego,
cada vez que detecte al jugador o las migas, se cambiarán de color y
luego se volverá al previo, de manera que resalte.*

```

164      rts
165 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
166 ; limpia_laberinto_especifico
167 ;   limpia el laberinto seleccionado sustituyendo por espacio
168 ;   si hay . o el jugador
169 ;   Entrada: X-direccion de comienzo de la cadena
170 ;   Salida: ninguna
171 ;   Registros afectados: CC.
172 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
173 limpia_laberinto_especifico:
174     ldd     temp
175     leax    d,x
176     clra
177     pshu    a,x
178     clrb
179 lolo:
180     tfr     x,y
181     lda     ,x+
182     beq     limpia_fin_laberinto_especifico
183     cmpa    #'X
184     bne     limpia
185     bra     lolo
186 limpia:
187     clra
188     lda     #32 ;Si encuentra algo distinto a X, pone un espacio en su lugar
189     sta     ,y
190     clra
191     bra     lolo
192 limpia_fin_laberinto_especifico:
193     pulu    a,x
194     rts
195 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

limpia_laberinto_especifico:

Borra un laberinto en concreto y lo deja en la forma estándar.

*El resto de subrutinas: **calculo_pos_inicial**, **calculo_pos_final**, **calculo_desplazamiento** y **calculos** se encuentran comentadas y totalmente detalladas paso por paso dentro del código fuente por lo que comentarlas aquí sería repetir lo mismo.*

*La última subrutina, **imprime_numeros**, va comparando la cifra con otros números, de manera que se reduce mientras se van imprimiendo sus cifras.*

NOTAS FINALES

El programa cumple con todos los requisitos especificados, y es capaz de funcionar con todo tipo de laberintos, tanto en tamaño como número (siempre dentro de las capacidades del ensamblador, es decir, máximo 255 laberintos porque es el que acaba en un registro.)

*Al ser un lenguaje tan vertical (pocas palabras por línea y muchas líneas) las capturas en este documento pueden ser dificultosas de leer, es mejor tenerlas como aproximación y ver el propio código fuente en el editor de textos. Hemos intentado plasmar las ideas principales aquí, pero por ejemplo las subrutinas de cálculos conllevaban muchas variables diferentes que extenderían demasiado este documento con las capturas, por eso recomendamos leerlo en el fuente **subrutinas.asm**, de igual manera pensamos que es más sencillo de esa manera. Los comentarios siempre irán ligados al fuente, este documento, en cambio podría perderse a lo largo del tiempo.*

Ha sido un trabajo muy costoso en el cual hemos aprendido más sobre el lenguaje base de los ordenadores y ha contribuido en el desarrollo de aprendizaje en la optimización de algoritmos dado las limitaciones que nos suponía el m6809.

Un saludo.