

Programa para descargar código objeto a través del puerto serial (downloader)

23 de octubre de 2025

M.I. Pedro Ignacio Rincón Gómez

ESTRUCTURA Y PROGRAMACIÓN DE
COMPUTADORAS

Programa para descargar código objeto a través del puerto serial (downloader)

Comúnmente los programas suelen grabarse en una memoria ROM, con el fin de conservarlos cuando se suspende el suministro eléctrico. Recordemos que un programa residente en RAM se pierde cuando la memoria no tiene suministro eléctrico.

Sin embargo, cuando se desarrolla un programa en lenguaje ensamblador el proceso de grabación y depuración del código puede ser costoso, tedioso y poco práctico.

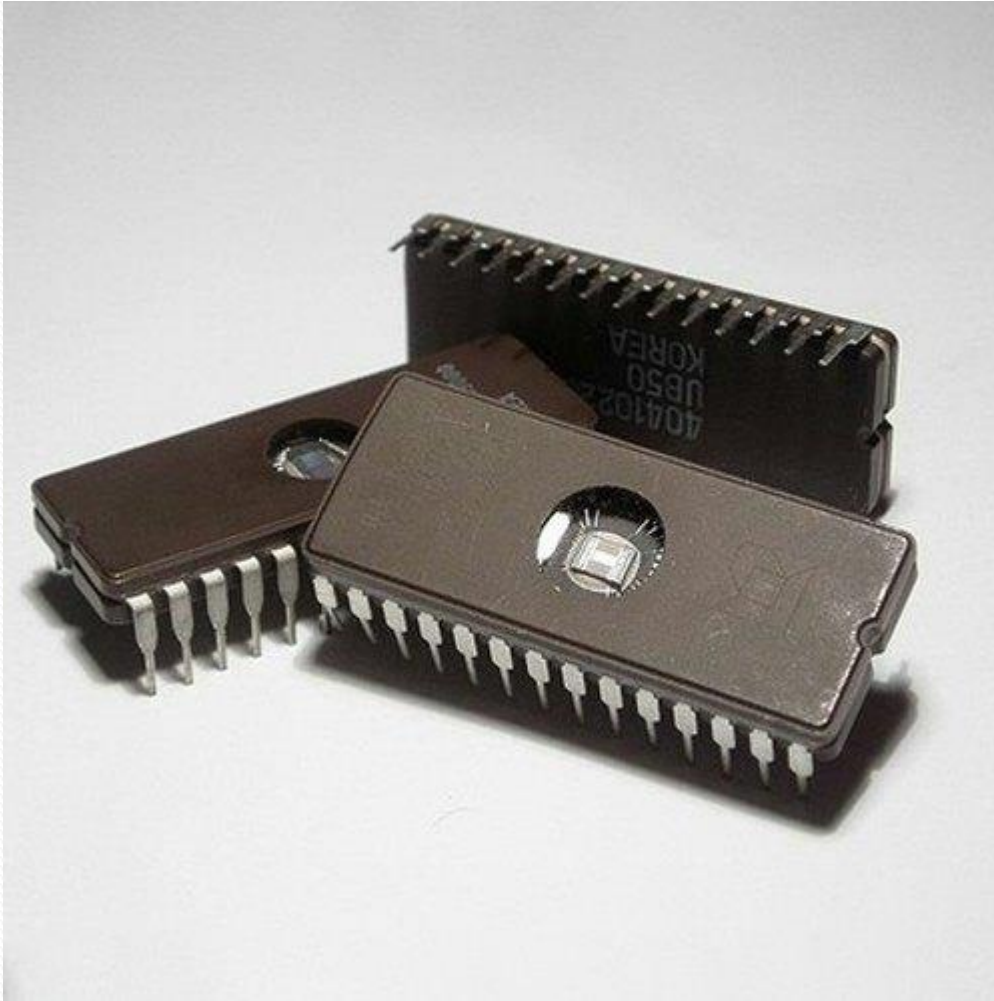
Recordemos que para grabar información en una memoria ROM se requiere de un grabador de memorias que suele ser bastante costoso.



Una vez que una memoria se graba, debe de probarse el funcionamiento del programa. Si éste no funciona como se esperaba, el dispositivo debe borrarse mediante la exposición a luz ultravioleta.

En la imagen de abajo se aprecia un borrador de memorias de luz ultravioleta. Dada la peligrosidad que implica la exposición de este tipo de luz a la vista, el borrador cuenta con un compartimiento donde se colocan las memorias que serán borradas.





Los dispositivos que son susceptibles de borrar mediante la exposición a luz ultravioleta, suelen presentar una pequeña ventana de vidrio que permite ver el interior del circuito integrado.

Además están elaborados mediante una tecnología conocida como Cmos, que se daña muy fácilmente al exponerse a electricidad estática.

Una forma más práctica para el desarrollo de programas consiste en descargarlos desde una computadora personal hacia un microcontrolador a través de un puerto serial, mediante un software conocido como downloader.

El programa se aloja en un área de memoria RAM disponible y si se desea borrar, simplemente se debe interrumpir el suministro eléctrico de la memoria RAM del microcontrolador.



Estrategia para programar un downloader.

Se utilizará el programa de START visto la semana pasada y sólo se modificará la parte correspondiente a EXITO, ya que se considerará que lo que se transmite después de la última T de START es una secuencia numérica correspondiente a un programa.

START- XX_0 - XX_1 - XX_2 - ... - XX_{n-1} - XX_n

Código objeto de un programa previamente compilado

Donde cada término XX es un byte del código objeto

START- XX_0 - XX_1 - XX_2 - ... - XX_{n-1} - XX_n

Código objeto de un programa previamente compilado

Cada término XX será guardado en memoria RAM a partir de una dirección arbitraria.

Para el programa que usaremos será \$0030

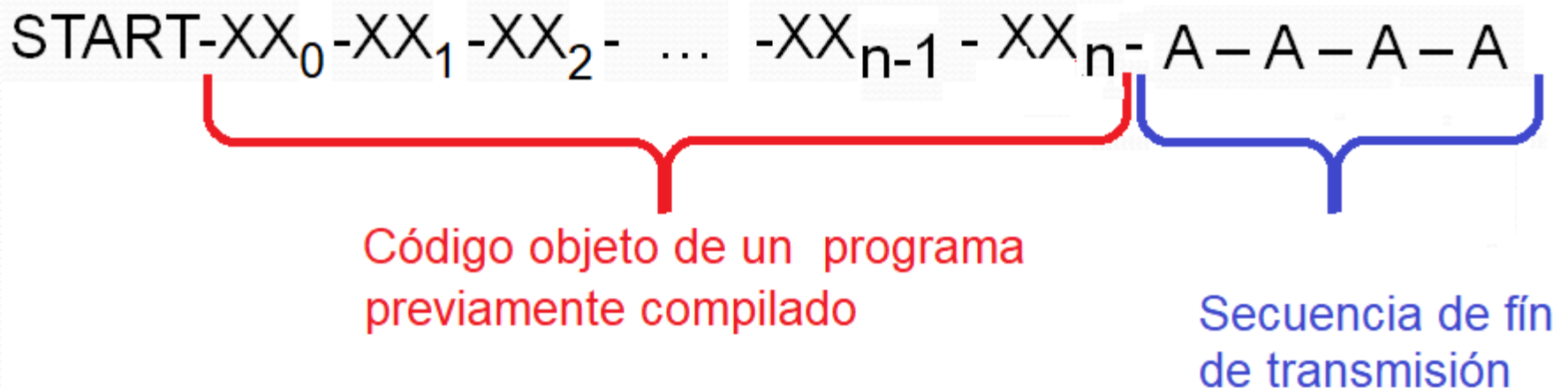
\$0030	XX_0
\$0031	XX_1
\$0032	XX_2
	\vdots
	XX_{n-1}
	XX_n

START- XX_0 - XX_1 - XX_2 - ... - XX_{n-1} - XX_n

Código objeto de un programa previamente compilado

Estrategia para programar un downloader.

Así como se cuenta con una cadena de caracteres que indica que se transmitirá un programa, se requiere de una secuencia de caracteres, para indicar el fin de la transmisión del mismo. Esta última secuencia es muy importante pues es la que indicará que ya se debe ejecutar el programa transmitido.



La cadena de fin de transmisión compuesta por cuatro caracteres A , obedece a las siguientes razones:

- 1) La cadena final debe estar compuesta por lo menos por cuatro elementos, ya que existen instrucciones en ensamblador que cuentan con tres operandos.

(Recordemos que los operandos pueden adquirir cualquier valor, inclusive el código ASCII de A) con cuatro elementos, se garantiza de que los caracteres no son simples operandos que forman parte de una instrucción.

- 
- 2) En el set de instrucciones del MC68HC11 no existe el código de instrucción número \$41 (Equivalente al código ASCII de la letra A)

Esto garantiza que nunca se podrán encontrar en un programa para el MC68HC11 a cuatro caracteres A consecutivos como parte del código objeto.

Consideraciones para programar un downloader.

Recordamos que en el programa START se inicializaba la variable ORDEN con un signo de interrogación “?” puesto que éste símbolo no forma parte de la secuencia START. En este programa no podemos usar esa estrategia, pues la secuencia numérica de un programa si puede incluir ese carácter.

Por tal motivo se utilizará una variable denominada VAR, que será inicializada en cero y que cada que tengamos una interrupción del puerto serial cambie a otro valor diferente.

Consideraciones para programar un downloader.

Adicionalmente utilizaremos una variable denominada `CONTA` que será inicializada con un valor de cero y tendrá la función de contar una secuencia de 4 caracteres "A" para terminar de recibir el programa.

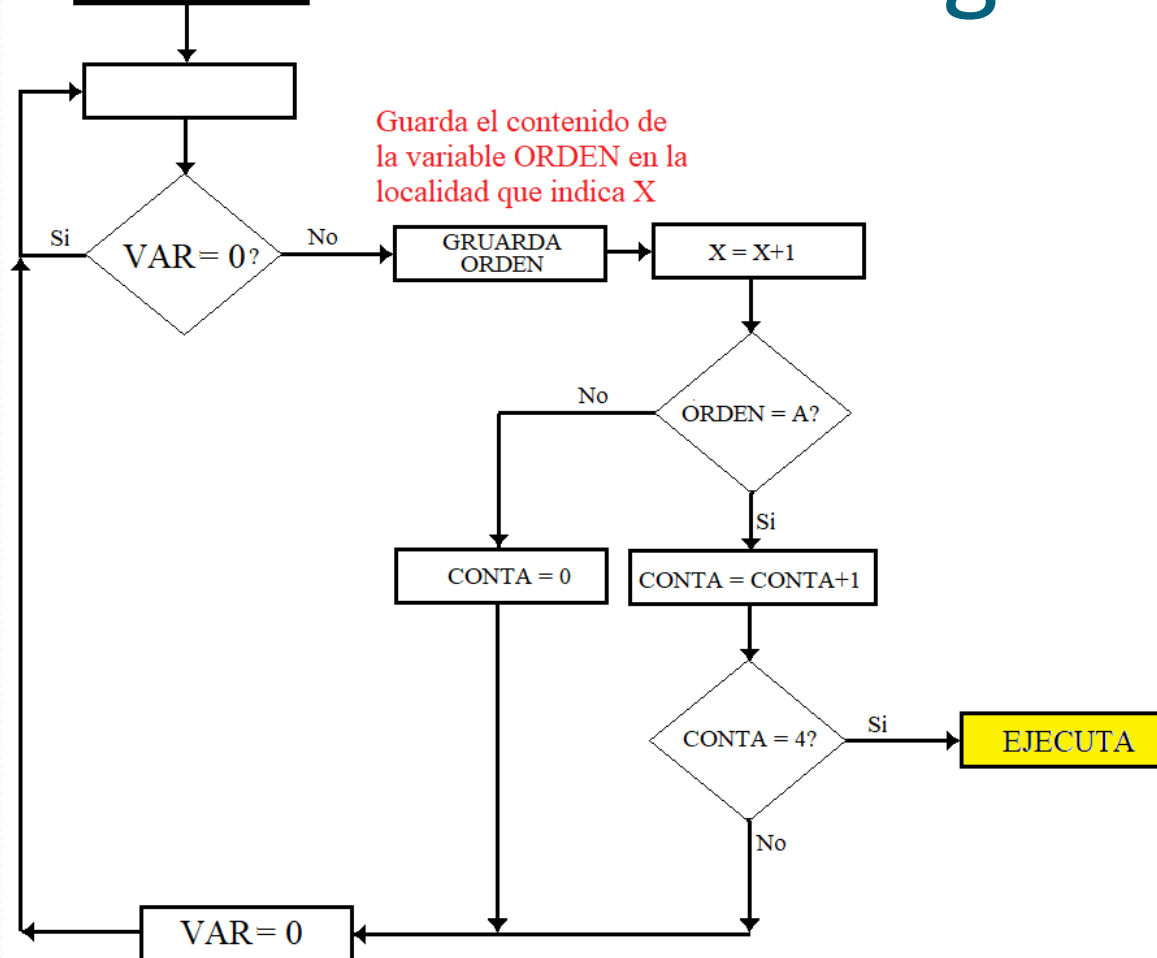
Si se presenta una, dos o tres caracteres de tipo A, pero el siguiente es diferente de A, entonces borrará por completo el contenido de `CONTA`.

Si por el contrario, se detectan en `CONTA` 4 caracteres "A", se procede a ejecutar el programa transmitido.

CONTA=0
ORDEN =0
VAR =0
X=DIR BASE

DIR_BASE es una constante que indica el origen del sector de memoria RAM donde se guardará el programa transmitido

Diagrama de flujo de EXITO



Para simular el programa downloader se requiere tener un programa de ejemplo, previamente compilado (Que será el programa a descargar)

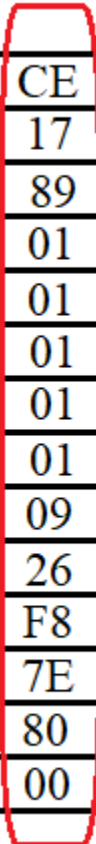
```
ORG $8000
INICIO
    LDX #$1789
LOOP
    NOP
    NOP
    NOP
    NOP
    NOP
    DEX
    BNE LOOP
    JMP INICIO
END
```

Compilando el programa se tiene:

```
                ORG $8000
                INICIO
8000 CE 17 89      LDX #$1789
                LOOP
8003 01           NOP
8004 01           NOP
8005 01           NOP
8006 01           NOP
8007 01           NOP
8008 09           DEX
8009 26 F8        BNE LOOP
800B 7E 80 00     JMP INICIO
                END
```

\$8000	CE
\$8001	17
\$8002	89
\$8003	01
\$8004	01
\$8005	01
\$8006	01
\$8007	01
\$8008	09
\$8009	26
\$800A	F8
\$800B	7E
\$800C	80
\$800D	00

La serie de números que se debe transmitir es:



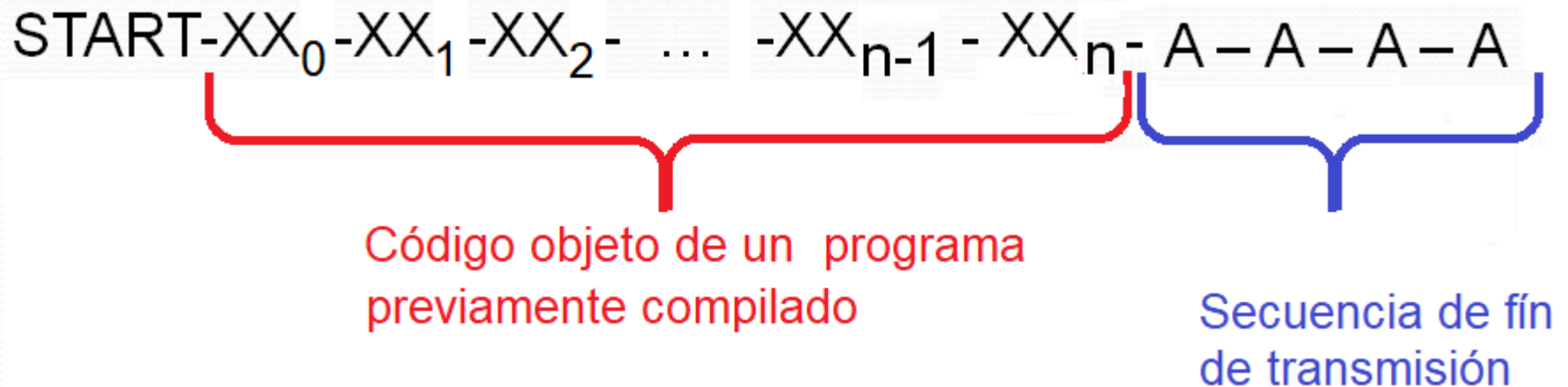
\$8000	CE
\$8001	17
\$8002	89
\$8003	01
\$8004	01
\$8005	01
\$8006	01
\$8007	01
\$8008	09
\$8009	26
\$800A	F8
\$800B	7E
\$800C	80
\$800D	00

El formato de números que soporta el simulador es decimal por lo que se convierte el contenido a su equivalente :

Hexadecimal		Decimal	
\$8000	CE	\$8000	206
\$8001	17	\$8001	023
\$8002	89	\$8002	137
\$8003	01	\$8003	001
\$8004	01	\$8004	001
\$8005	01	\$8005	001
\$8006	01	\$8006	001
\$8007	01	\$8007	001
\$8008	09	\$8008	009
\$8009	26	\$8009	038
\$800A	F8	\$800A	248
\$800B	7E	\$800B	126
\$800C	80	\$800C	128
\$800D	00	\$800D	000

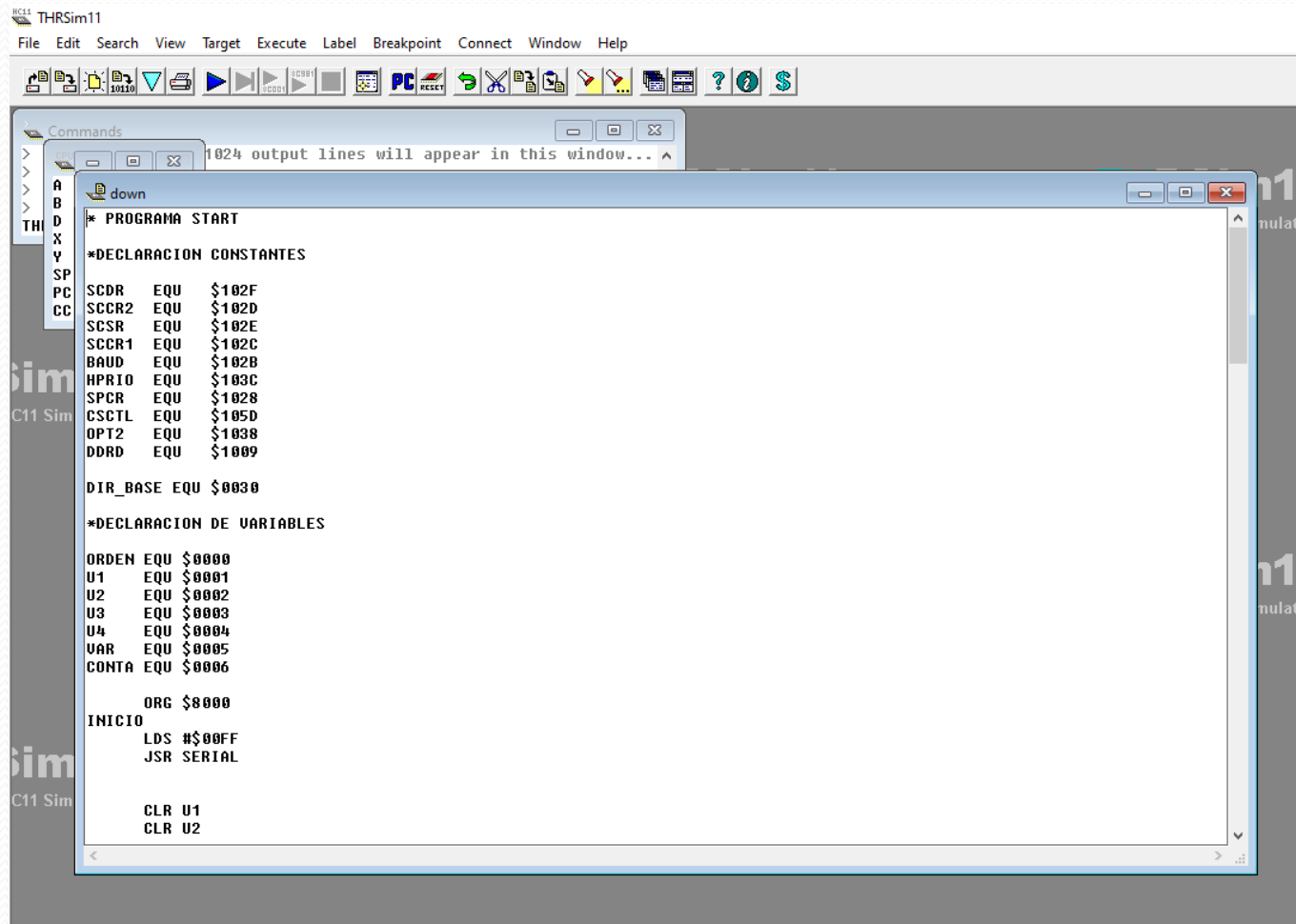
Esta es la cadena que se debe introducir en la ventana de “Serial Transmitter”

START<206><023><137><001><001><001><001><001><009><038><248><126><128><000>AAAA

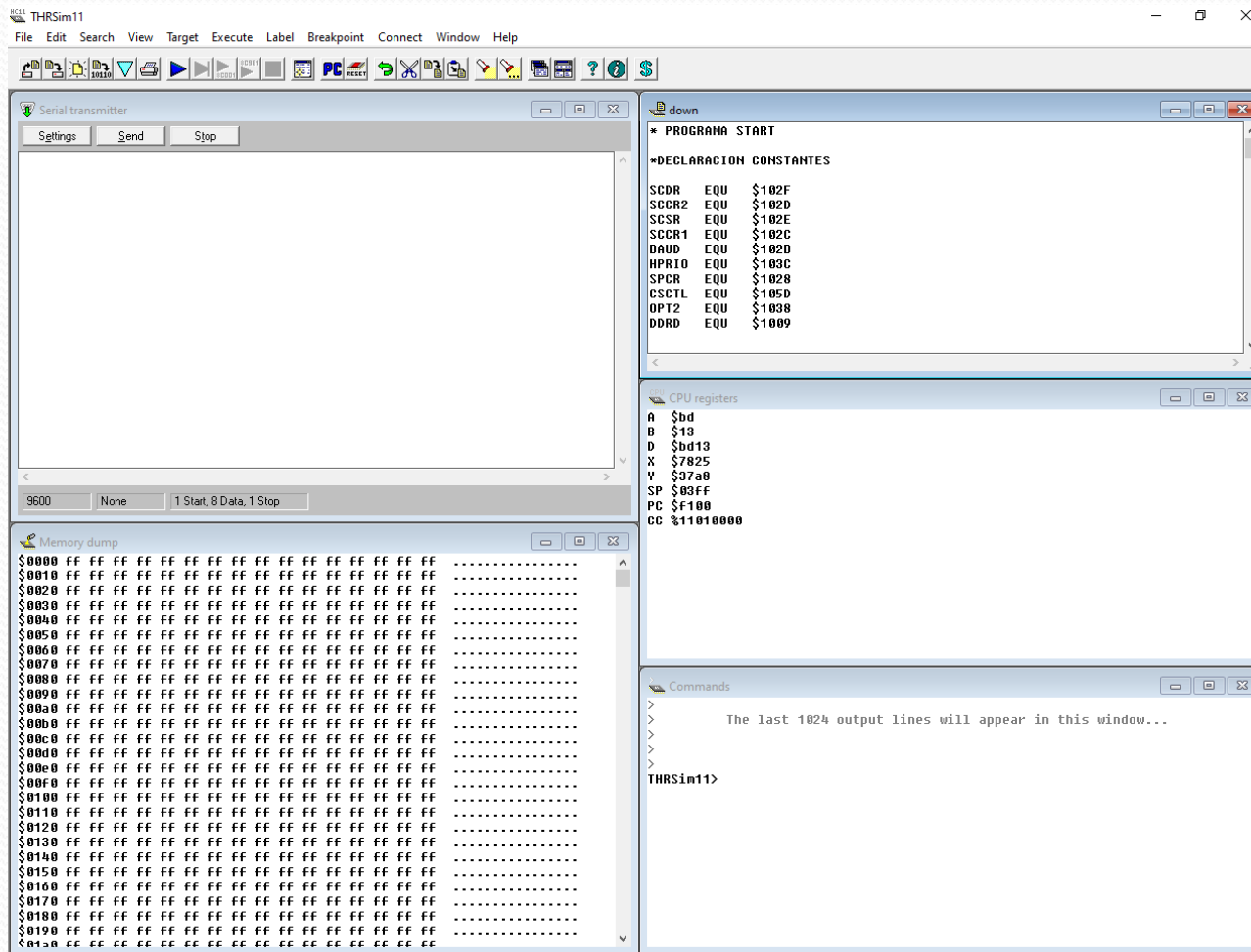


Es de suma importancia poner los símbolos <> para cada cifra en decimal

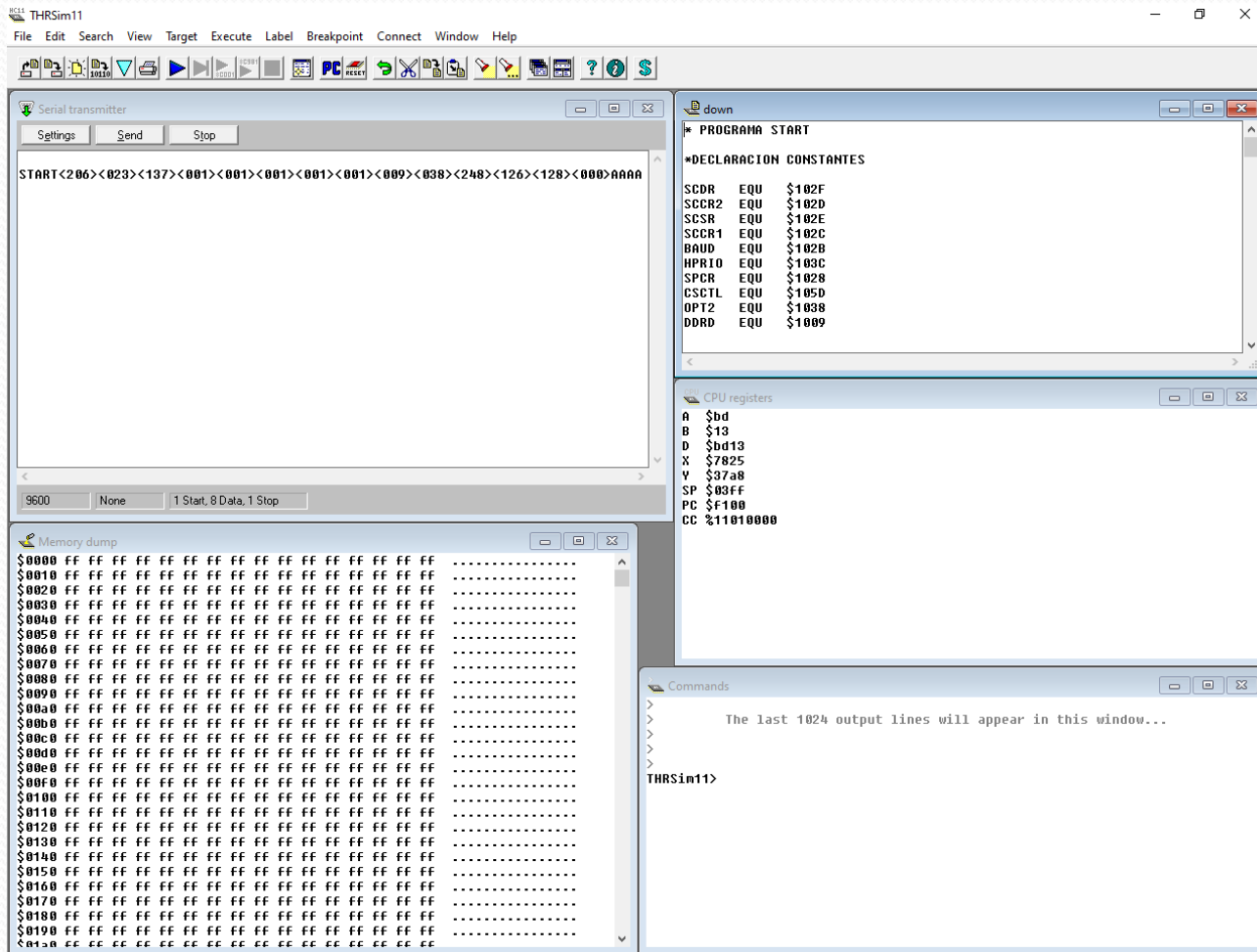
Proceso de simulación: Paso 1 cargar el programa down.asc en el simulador THRSIM11



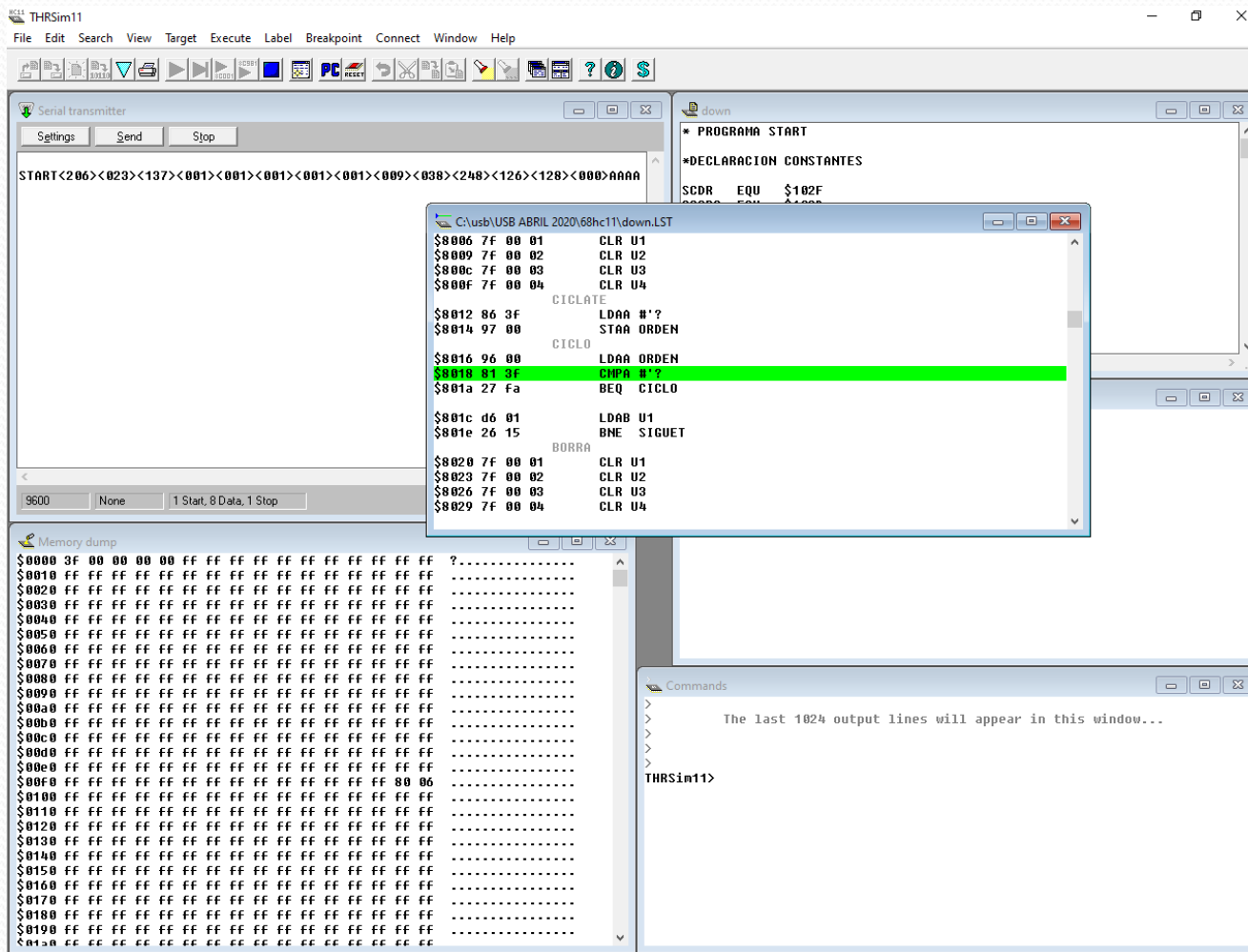
Proceso de simulación: Paso 2 abrir ventanas de DUMP (a partir de la dirección \$0000) y Serial Transmitter



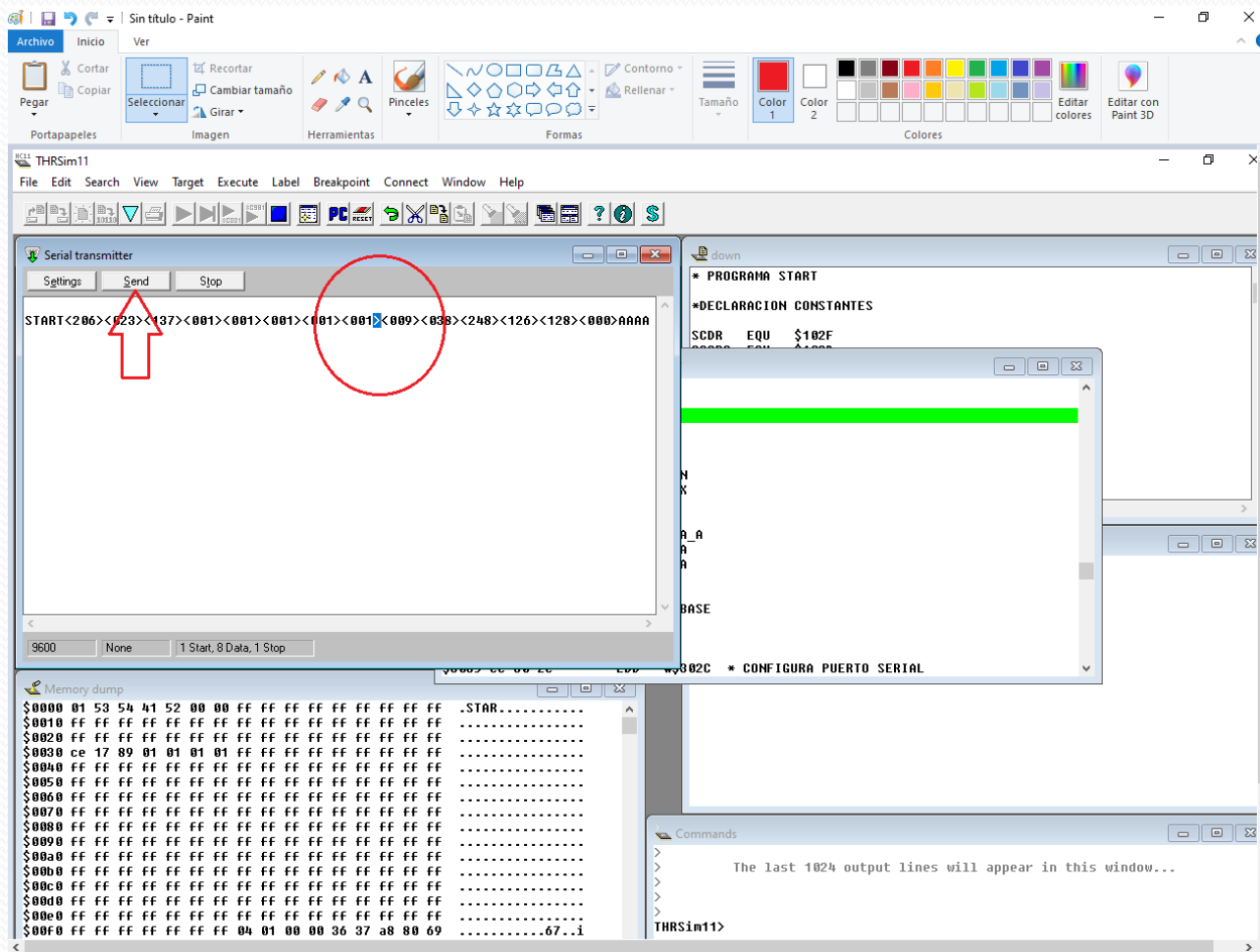
Proceso de simulación: Paso 3 escribir la cadena a transmitir en la ventana “Serial Transmitter”



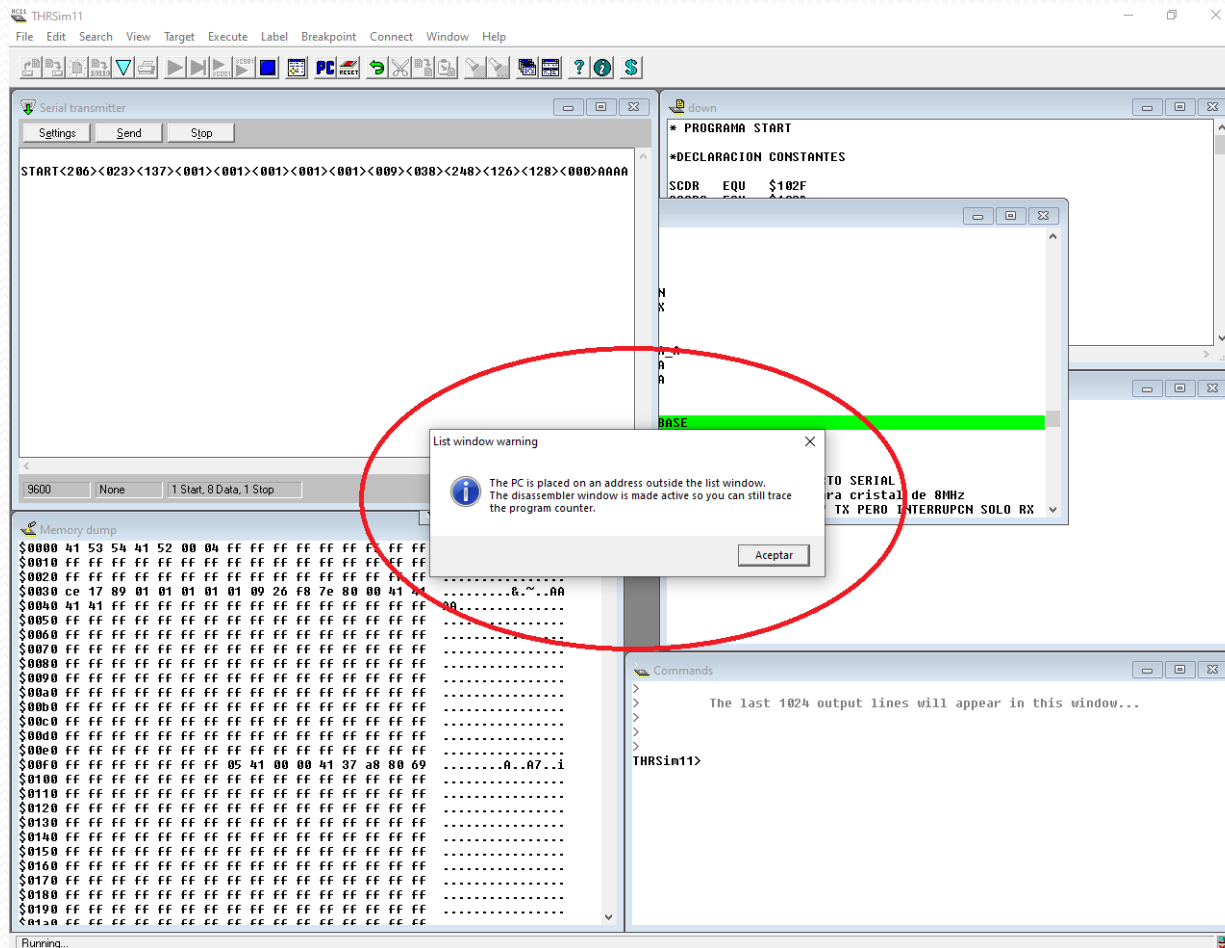
Proceso de simulación: Paso 4 ejecutar el programa down.asc



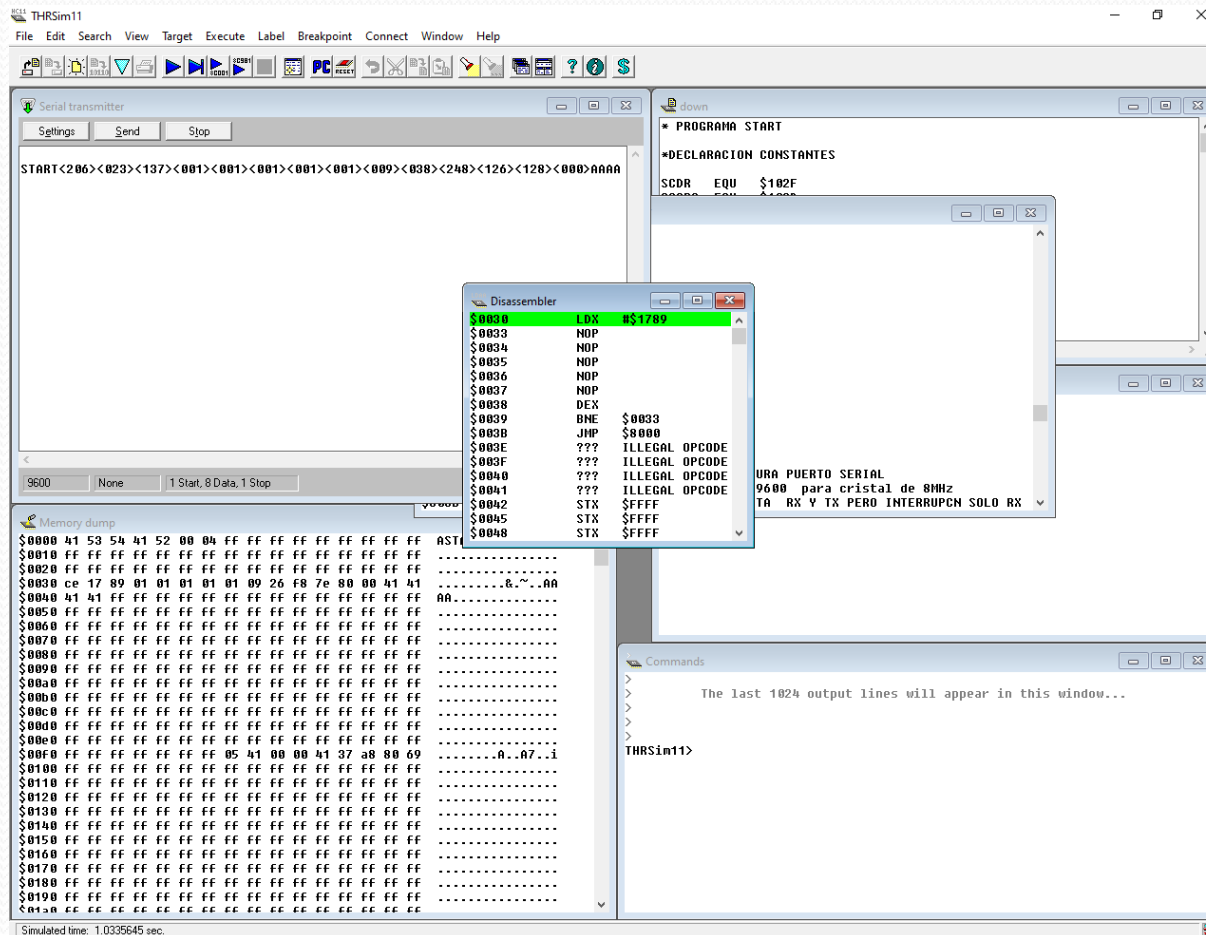
Proceso de simulación: Paso 5 oprimir el botón “Send” para transmitir la cadena.



Proceso de simulación: Paso 6 oprimir el botón “aceptar” en la ventana de warning.



Proceso de simulación: Paso 7 puede observarse que el programa transmitido se ejecuta en RAM



EJERCICIOS

- 1) Compare el diagrama de flujo que aparece en esta presentación, con el código del programa denominado “down.asc” que se anexó en el correo electrónico.
- 2) Simule el programa down.asc haciendo uso del simulador visto en clase THRsim11.exe. (Para ello es necesario simular la transmisión de una cadena de caracteres, a través de la función “Serial transfer” que se encuentra en el menú de memoria. Y describa el comportamiento del programa brevemente.)
- 3) Modifique la secuencia del programa que se transmite de modo que se quede permanentemente funcionando en RAM.