

Práctica No. 1

Introducción al intérprete 80X86 sobre la plataforma T-Juino

Objetivo: El alumno se familiarizará con el intérprete 80X86 (modo 16bits) sobre la plataforma T-Juino. Esto con el fin de desarrollar programas en lenguaje ensamblador para dicha plataforma.

Material:

- Computadora Personal (PC)
- Programa Edito de texto (ASCII), TASM y TLINK
- Tarjeta T-Juino (con intérprete 80x86)
- Manejador FTDI instalado

Equipo:

- Computadora Personal
- Programa emulador de terminal

Introducción

La tarjeta T-Juino es una plataforma de hardware basada en un PCB con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de microcontroladores en proyectos multidisciplinarios.

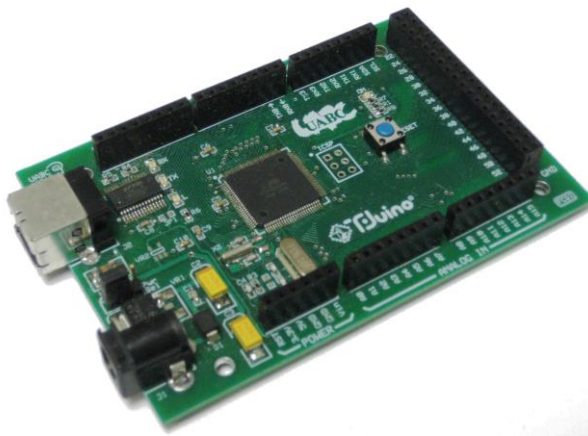


Figura 1. Tarjeta T-Juino.

T-Juino fue diseñado para ser compatible con la plataforma libre llamada Arduino (específicamente Arduino Mega) y el sistema SM8088 de UABC. El hardware consiste de una placa con un microcontrolador Atmel AVR con sus puertos de entrada/salida expuesto mediante conectores. El microcontrolador utilizado en el diseño es el Atmega1280 y para el desarrollo de software se puede utilizar entornos de desarrollo libres como Arduino IDE, Wiring o WinAVR+ Programmer notepad.

Es importante remarcar que una diferencia con la plataforma Arduino es que en T-Juino se cuenta con una máquina virtual (VM88) del procesador 80x86 de 16 bits desarrollada en UABC. Esta máquina lo hace compatible con los programas ejecutables que fueron desarrollados para la plataforma SM8088 basada en el procesador clásico de Intel 80C88.

Desarrollo:

- Conecte la tarjeta T-Juino a la computadora personal (PC) mediante el cable USB conecte y asegúrese que haya sido reconocido como un puerto serie.
- Ejecute un programa emulador de terminal (p.e. MTTY) y configure la comunicación serie a **19200,8,N,1** (Velocidad 19200 Baud, 8 bits de datos, No paridad y 1 bit de paro.).
- Seleccione el puerto serie a utilizar correspondiente al puerto donde haya sido reconocido T-Juino.
- Una vez establecidos todos los parámetros proceda a realizar la comunicación mediante el botón de “Connect”.
- Si la comunicación es exitosa aparecerá el símbolo de interrogación junto con el de mayor que (?>). En ese comentario presione la tecla ENTER para entrar en modo “monitor”. **Nota:** Puede presionar el botón de RESET de T-Juino para intentarlo de nuevo.
- El modo monitor presenta la pantalla de inicio mostrada en Figura 2 y estando en ese modo se puede interactuar con el intérprete.

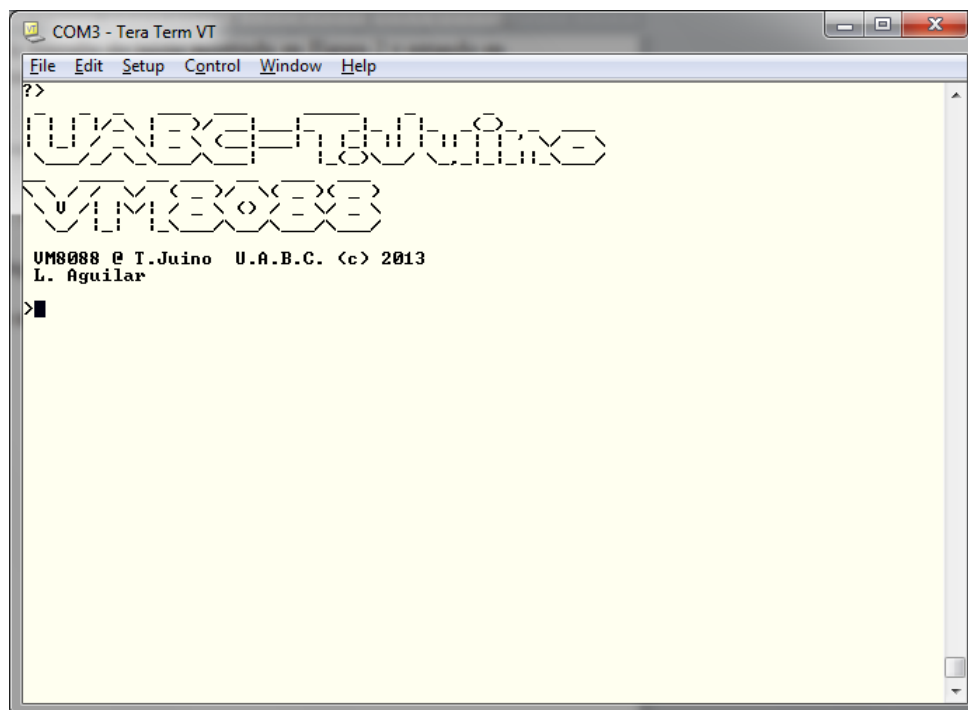


Figura 2. Pantalla del mensaje de inicio del VM8088 y el comando r (registers)

Los comando aceptados en este modo se pueden mostrar mediante la tecla ? que corresponde al comando de **help**.

Nota: no todos los comandos desplegados están implementados y algunos se encuentra en proceso de su implementación.

Los comandos del modo monitor se muestran a continuación.

```
*clear      C <range>
dump        D <range>
*enter      E address <list>
*fill       F range list
go          G
help        ?
input       I port
load        L
*move       M range address
output      O port value
ports       P
quit        Q
register     R <register value>
trace       T
unassemble  U <range>
version     V
zero (rst)  Z
* - in process
```

Comandos

d – *dump*: hace un desplegado del contenido de un rango de memoria, el desplegado se hace en formato decimal y ASCII (si el contenido es desplegable).

g – *go*: ejecuta el programa que se encuentra cargado en la memoria a partir de la dirección 100h.

? – *help*: muestra la lista de comandos permitidos en el modo monitor.

i – *input*: leer un byte de un puerto dado.

l – *load*: carga en memoria un archivo (programa) a partir de la dirección 100h.

o – *output*: escribe un byte a un puerto dado.

p – *ports*: muestra el valor actual de los puertos de entrada-salida (E/S).

q – *quit*: sale del modo monitor e inicia el modo de ejecución.

r – *registers*: muestra el valor actual de todos los registros del procesador, además puede ser usado para modificar los registro del procesar.

t – *trace*: ejecutar la instrucción apuntado por el CS:IP y muestra los valores resultantes de los registros.

u – *unassemble*: desensambla un rango de memoria dado mostrando los mnemónicos correspondiente.

v – *version*: muestra la versión del interprete que se encuentra cargada en T-Juino.

z – *reset*: realiza un reset sobre el intérprete inicializando todos sus registros en cero excepto IP que lo inicializa en 100h.

- 1) Basándose en listado 1 crear in archivo texto llamado **pra1.asm**.

Listado 1

```
.model tiny

;----- Insert INCLUDE "filename" directives here
;----- Insert EQU and = equates here

locals

.data
    Mens DB 'Hola Mundo',10,13,0

.code
    org 100h

;*****
; Procedimiento principal
;*****
principal proc
    mov     sp,0ffffh        ; inicializar SP (Stack Pointer)
    @@ini0: mov     dx,1
    @@ini1: mov     cx,dx
    @@sigue: mov     al,'x'
             call    putchar
             loop    @@sigue
             mov     al,10
             call    putchar
             mov     al,13
             call    putchar
             inc     dx
             cmp     dx,20
             jbe     @@ini1
             jmp     @@ini0
             ret
    endp

;*****
; procedimientos
;*****
putchar proc
    push    ax
    push    dx
    mov     dl,al
    mov     ah,2             ; imprimir caracter DL
                             ; usando servicio 2 (ah=2)
    int     21h
    pop     dx               ; del int 21h
    pop     ax
    ret
    endp

end     principal          ; fin de programa (file)
```

- 2) Descargue los programas **tasm.exe**, **tlink.exe** de moodle (curso Microprocesadores y Microcontroladores en <http://uabc-io.net>) y deposítelos en un directorio exclusivo de trabajo (por ejemplo. C:\uPuC\).

- 3) Ensamble el programa **prac1.asm** mediante la línea de comando:

C:\uPuC>tasm prac1.asm

Esto generará el archivo **prac1.obj**

- 4) Encadena el archivo **obj** para generar programa **.com** mediante la línea de comando:

C:\uPuC> tlink /t prac1.obj

Esto generará el archivo ejecutable **prac1.com**

- Nota:** Este programa puede ser renombrado a **prac1.bin** con el fin de no ejecutarlo en la computadora.
- 5) Cargar y ejecutar el programa **prac1** en T-Juino.



```
COM3 - Tera Term VT
File Edit Setup Control Window Help
xxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxx
x
xx
xxx
xxxx
xxxxx
xxxxxx
xxxxxxx
xxxxxxx
xxxxxxx
xxxxxxx
xxxxxxx
xxxxxxx
xxxxxxx
xxxxxxx
xxxxxxx
xxxxxxx
xxxxxxx
xxxxxxx
x
xx
xxx
xxxx
xxxxx
xxxxxx
xxxxxxx
xxxxxxx
xxxxxxx
xxxxxxx
xxxxxxx
xxxxxxx
xxxxxxx
xxxxxxx
xxxxxxx
xxxxxxx
xxxxxxx
xxxxxxx
```

Figura 3. Pantalla de salida al ejecutar **prac1.com** con el intérprete en T-Juino.

Actividades a realizar:

1. **Elaborar un diagrama de flujo para ejecutar un programa sobre la tarjeta T-Juino (incluyendo el caso en el que no cuenta con el interprete del 8088).**
2. **Implementar un programa que retorna el n -ésimo “número feo” ([Números Regulares/Hamming](#)).**
Definiremos como número feo todos los enteros positivos que sus factores primos **solo** incluyen [2, 3, 5].

Ejemplos:

$$6 = 2 \cdot 3 \quad \therefore \text{Es feo}$$

$$8 = 2 \cdot 2 \cdot 2 \quad \therefore \text{Es feo}$$

$$14 = 7 \cdot 2 \quad \therefore \text{No es feo}$$

Nota: El número 1 generalmente se considera como un número feo.

Si generamos la secuencia de números feos: (1, 2, 3, 4, 5, 6, 8, 9, 10, 12), entonces el **décimo** número feo es el **12**.

Conclusiones y Comentarios.