

# MEMORIA

---

## ARQUITECTURA DE COMPUTADORES

**Pablo de Vicente Abad a180470**  
**Julia Serrano Arrese c200119**  
**2021-2022**

# ÍNDICE

[INTRODUCCIÓN](#)

[DIAGRAMA DE FLUJO](#)

[SUBROUTINAS](#)

[LEECAR](#)

[ESCCAR](#)

[INIT](#)

[SCAN](#)

[PRINT](#)

[RTI](#)

[CASOS DE PRUEBA](#)

[VALORACIONES FINALES](#)

# INTRODUCCIÓN

El proyecto consiste en controlar el manejo de operaciones de Entrada/Salida en un periférico mediante interrupciones. El dispositivo elegido es la DUART MC68681 operando ambas líneas mediante interrupciones. En el computador del proyecto la DUART está conectada a la línea de petición de interrupción de nivel 4.

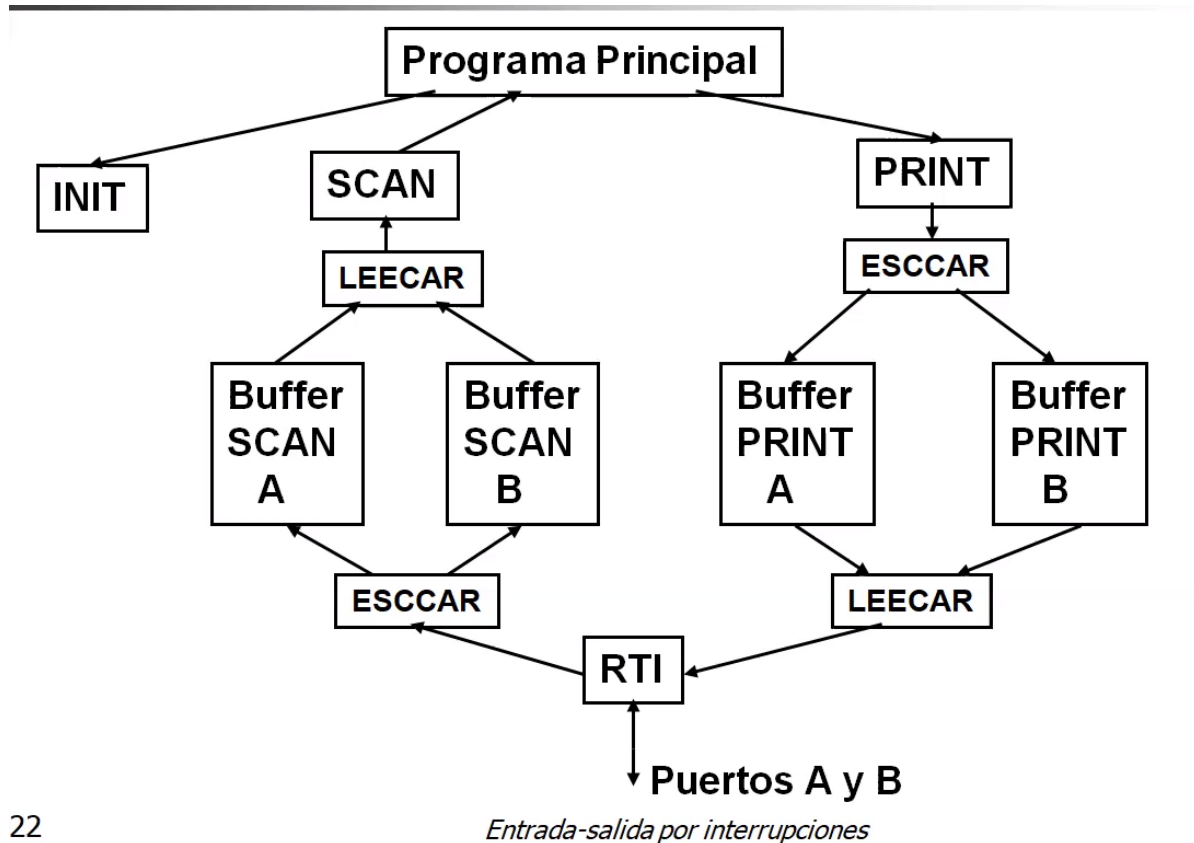
Para la elaboración del proyecto se necesitan unos búfferes internos para almacenar los caracteres que se reciben asíncronamente por las líneas. Del mismo modo, se necesitan sendos búfferes internos para almacenar los caracteres pendientes de transmitirse por las líneas. Además, existe una única rutina de tratamiento de las interrupciones de las líneas que será la encargada de transferir la información a/o desde los mencionados búfferes internos.

Las subrutinas a desarrollar son: INIT, SCAN, PRINT y RTI.

Como ayuda para la implementación y prueba de los búfferes internos que deben manipular las subrutinas anteriores, contamos con las siguientes subrutinas auxiliares que nos han proporcionado en los computadores asignados al proyecto (fichero bib\_aux.s): LEECAR, ESCCAR e INI\_BUFS.

Explicaremos el funcionamiento de las subrutinas en la sección SUBROUTINAS.

# DIAGRAMA DE FLUJO



Una vez identificada se debe realizar su tratamiento:

RxRDY se colocará a 1 si se ha recibido un carácter, entonces comenzamos con el procedimiento de recepción:

#### Procedimiento Recepción

Cuando llega un carácter a la línea, se genera una interrupción de recepción (FIFO de recepción no vacía). Se recoge el carácter de la línea y se realiza una llamada a ESCCAR. Este inserta el carácter en el buffer interno de la línea correspondiente de SCAN.

Cuando el programa principal llama a SCAN, este explora si hay caracteres en el buffer interno de la línea que se le haya pasado por el 'Descriptor' y lee la cantidad de caracteres que se le pasa en el parámetro 'Tamaño'.

#### Procedimiento Transmisión

El programa principal llama a PRINT. Print llama a ESCCAR 'tamaño' veces e inserta los caracteres en el buffer interno de PRINT. Como el buffer no está vacío, se habilitan las interrupciones de transmisión por la línea correspondiente. El dispositivo está preparado para transmitir por lo que se genera una interrupción de transmisión desde la RTI.

La RTI trata la interrupción de transmisión llamando a LEECAR, que obtiene el carácter del buffer interno correspondiente de PRINT y lo transmite por la línea correspondiente. Si el buffer interno está vacío se inhiben las interrupciones de transmisión por la línea actualizando el IMR.

# SUBROUTINAS

En esta sección explicaremos el funcionamiento de las subrutinas desarrolladas y las dadas en el fichero auxiliar.

De las subrutinas ya dadas explicaremos el funcionamiento general y los parámetros necesarios para su correcto funcionamiento:

## LEECAR

### Parámetros de entrada:

Posee un único parámetro de entrada:

- **Buffer**, de 4 bytes. Es un descriptor que indica de qué buffer interno de los 4 disponibles se desea extraer el carácter. Se pasa por D0.

Los posibles valores de los dos bits menos significativos del parámetro de entrada son:

- 0: buffer interno de recepción de la línea A.
- 1: buffer interno de recepción de la línea B.
- 2: buffer interno de transmisión de la línea A.
- 3: buffer interno de transmisión de la línea B.

### Parámetros de salida:

Posee un único parámetro de salida:

- **D0**, de 4 bytes. Se devuelve un código que indica el resultado de la operación:
  - D0 = 0xFFFFFFFF si no hay ningún carácter disponible en el buffer interno seleccionado.
  - D0 es un número entre 0 y 255 que indicará el carácter que se ha extraído del buffer interno seleccionado.

### Funcionamiento:

La rutina LEECAR realiza la extracción de un carácter del buffer interno que se selecciona en el parámetro.

# ESCCAR

## Parámetros de entrada:

Posee dos parámetros de entrada:

- **Buffer**, de 4 bytes. Es un descriptor que indica de qué buffer interno de los 4 disponibles se desea obtener el carácter. Se pasa por D0.

Los posibles valores de los dos bits menos significativos del parámetro de entrada son:

- 0: buffer interno de recepción de la línea A.
  - 1: buffer interno de recepción de la línea B.
  - 2: buffer interno de transmisión de la línea A.
  - 3: buffer interno de transmisión de la línea B.
- **Caracter**, de 1 byte. Es el carácter que se desea añadir al buffer interno. Se pasa por valor en los 8 bits menos significativos del registro D1.

## Parámetros de salida:

Posee un único parámetro de salida:

- **D0**, de 4 bytes. Se devuelve un código que indica el resultado de la operación:
  - D0 = 0xFFFFFFFF si el buffer interno seleccionado está lleno.
  - D0 = 0 si el carácter se ha insertado en el buffer interno correctamente.

## Funcionamiento:

La rutina ESCCAR realiza la inserción de un carácter del buffer interno que se selecciona en el parámetro

A continuación, explicamos el resto de subrutinas pedidas con más detalle ya que son las que hemos realizado nosotros:

## INIT

No tiene parámetros de entrada ni de salida.

### Funcionamiento:

Se encarga de la inicialización de los dispositivos. Prepara las líneas A y B para la transmisión y recepción de caracteres mediante E/S por interrupciones. Se llama a la subrutina INI\_BUFS proporcionada en el código auxiliar para realizar la inicialización de los cuatro búfferes internos.

Los parámetros de inicialización de esta subrutina son: 8 bits por carácter para ambas líneas, no activar eco en ninguna de las líneas, solicitar una interrupción cada vez que llegue un caracter, velocidad de transmisión y recepción de 38400 bits/s, funcionamiento Full Duplex, vector de interrupción \$40, habilitar las interrupciones de recepción de las líneas en la máscara de interrupción, actualizar la dirección de la rutina de tratamiento de interrupción en la tabla de vectores de interrupción y por último no olvidar realizar la copia de IMR (llamada IMR\_2 en nuestro caso) dado que no se puede leer de este valor.

Las subrutinas que vamos a explicar ahora reciben los parámetros en la pila y el valor de retorno se devuelve en el registro D0.

## SCAN

### Parámetros de entrada:

Posee dos parámetros de entrada:

- **Descriptor**, de 2 bytes .Es un número entero. Indica el dispositivo sobre el que se desea realizar la operación de lectura:
  - 0 indica que la lectura se realizará de la línea A.
  - 1 indica que la lectura se realizará de la línea B.
  - Cualquier otro valor provocará que la subrutina devuelva un error.



- **Tamaño**, de 2 bytes. Es un número entero sin signo. Indica el número máximo de caracteres que se deben leer del buffer interno y copiar en el parámetro Buffer.

#### Parámetros de salida:

Posee un único parámetro de salida:

- **D0**, de 4 bytes. Se devuelve un código que indica el resultado de la operación:
  - D0 = 0xFFFFFFFF si existe algún error en los parámetros pasados.
  - D0 es un número positivo. Indicará el número de caracteres que se han leído y se han copiado a partir de la posición de memoria indicada por el parámetro 'Buffer'.
- **Buffer**, de 4 bytes. Es el buffer en el que se van a devolver los caracteres que se han leído del dispositivo.

#### Funcionamiento:

La rutina SCAN realiza la lectura de un bloque de caracteres de la línea correspondiente (A o B).

Copia en el parámetro 'Buffer' los 'Tamaño' primeros caracteres almacenados en el buffer interno correspondiente y los "elimina" de dicho buffer interno utilizando la función LEECAR.

Aspectos a tener en cuenta:

- La lectura se deberá realizar de forma no bloqueante
- Si el buffer interno contiene menos de Tamaño caracteres, los copiará en el parámetro Buffer y el buffer interno pasará a estar vacío.
- En D0 se almacenará el número de caracteres que se han copiado en Buffer.
- Esta subrutina debe asegurar que ningún carácter es leído dos veces y que no se pierde ninguno
- Se copiarán a lo sumo tantos bytes como indique el parámetro Tamaño.

### Nuestro código:

Tras cargar los parámetros en la pila, chequeamos que el 'Tamaño' y el 'Descriptor' son adecuados, en caso contrario se salta a la etiqueta llamada ERROR\_SCAN que devuelve 0xFFFFFFFF en D0.

El 'Descriptor' nos muestra la línea sobre la que se realizará la lectura. Hemos creado 2 etiquetas en función de la línea (ambas siguen la misma estructura): ESP\_SCAN y ESPSCANB.

Se tratan de bucles de copia de caracteres en el parámetro 'Buffer', "eliminando" los caracteres de los buffers internos a través de la llamada a la subrutina LEECAR.

Cuando ya no quedan más caracteres (ya se han copiado los 'Tamaño' caracteres) o el buffer interno está vacío, se sale del bucle y salta a la etiqueta FIN\_SCAN. Movemos nuestro auxiliar a D0 y restauramos los registros utilizados.

## PRINT

### Parámetros de entrada:

Posee tres parámetros de entrada:

- **Buffer**, de 4 bytes. Es el buffer en el que se pasa el conjunto de caracteres que se desea escribir en el dispositivo
- **Descriptor**, de 2 bytes. Es un número entero. Es un parámetro de entrada. Indica el dispositivo sobre el que se desea realizar la operación de escritura:
  - 0 indica que la escritura se realizará de la línea A.
  - 1 indica que la escritura se realizará de la línea B.
  - Cualquier otro valor provocará que la subrutina devuelva un error.
- **Tamaño**, de 2 bytes. Es un número entero sin signo. Indica el número de caracteres que se deben leer del parámetro Buffer y escribir en el puerto.

### Parámetros de salida:

Posee un único parámetro de salida:

- **D0**, de 4 bytes. Se devuelve un código que indica el resultado de la operación:
  - D0 = 0xFFFFFFFF si existe algún error en los parámetros pasados.
  - D0 es un número positivo. Indicará el número de caracteres que se han aceptado para su escritura en el dispositivo.

#### Funcionamiento:

Escritura en un dispositivo. Ordenará la escritura de un bloque de caracteres por la línea correspondiente (A o B).

La rutina PRINT realiza la escritura en el correspondiente buffer interno de tantos caracteres como indique el parámetro 'Tamaño' contenidos en el 'buffer' que se pasa como parámetro.

Si como resultado de dicha copia hay caracteres en el buffer interno, se activa la transmisión de caracteres por la línea.

La copia de los caracteres se realiza invocando a la función ESCCAR.

#### Aspectos a tener en cuenta:

- La escritura se deberá realizar de forma no bloqueante
- El MC68681 generará una interrupción cuando alguna de las líneas esté preparada para transmitir
- La rutina de tratamiento de interrupción será la encargada de ir transmitiendo los caracteres por la línea correspondiente
- En D0 se almacenará el número de caracteres que se han copiado en Buffer.
- En el caso de que los caracteres que se desean transmitir no quepan en su totalidad en el buffer interno, se "encolan" los que caben y se devuelve el número de caracteres copiados en D0.
- La DUART solicita interrupciones cada vez que la línea correspondiente esté lista para transmitir si se han activado en el registro de máscara de interrupciones (IMR).

#### Nuestro código:

Tras cargar los parámetros en la pila, chequeamos que el 'Tamaño' y el 'Descriptor' son adecuados, en caso contrario se salta a la etiqueta llamada ERROR\_PRINT que devuelve 0xFFFFFFFF en D0.

El 'Descriptor' nos muestra la línea sobre la que se realizará la escritura. Hemos creado 2 etiquetas en función de la línea (ambas siguen la misma estructura): ESP\_PRINTA y ESPPRINTB.

Se tratan de bucles de copia de caracteres del parámetro 'Buffer' al buffer interno a través de la llamada a ESCCAR.

Si como resultado de dicha copia hay caracteres en el buffer interno, se activa la transmisión de caracteres por la línea, solicitando una interrupción (se activa el bit adecuado de IMR a través de su copia, IMR\_2).

Cuando ya no quedan más caracteres (ya se han copiado los 'Tamaño' caracteres) o el buffer interno está lleno, se sale del bucle y salta a la etiqueta FIN\_PRINT. Movemos nuestro auxiliar a D0 y restauramos los registros utilizados.

# RTI

No tiene parámetros de entrada ni de salida.

## Funcionamiento:

La invocación de la rutina de tratamiento de interrupción es el resultado de la ejecución de la secuencia de reconocimiento de interrupciones

Entre otras acciones esta subrutina debe realizar las siguientes acciones:

- Identificación de la fuente de interrupción. Esta subrutina debe identificar cuál de las cuatro posibles condiciones ha generado la solicitud de interrupción.
- Tratamiento de la interrupción:
  - Interrupción de recepción: indica que la cola FIFO de recepción de la línea no está vacía. Se añade el carácter que se recibió por la línea al buffer interno correspondiente utilizando la función ESCCAR.
  - Interrupción de transmisión: indica que la línea está preparada para transmitir un carácter. Si quedan caracteres en el buffer interno de transmisión, se debe obtener el primer carácter del buffer interno, “eliminarlo” invocando a la función LEECAR y transmitirlo por la línea.

## Nuestro código:

Lo primero que hacemos es identificar la fuente de interrupción, si no hay interrupción se finaliza.

Una vez identificada la fuente de interrupción tratamos la interrupción, para ello hemos creado 4 etiquetas, 2 de recepción y 2 de transmisión: REC\_A, REC\_B, TRA\_A y TRA\_B.

Las interrupciones de recepción indican que la cola FIFO de recepción no está vacía, por lo tanto cargamos los parámetros de entrada necesarios para llamar a ESCCAR que añade el carácter recibido al buffer interno correspondiente (si es REC\_A lo añade al buffer A de recepción y REC\_B al buffer B de recepción).

Cuando el buffer interno está lleno, salta a FIN\_RTI, donde se restauran los registros utilizados, en caso contrario vuelve a la sección de identificación de la fuente de interrupción (ESP\_RTI).

Las interrupciones de transmisión indican que la línea está preparada para transmitir un carácter, por lo tanto, cargamos el parámetro de entrada necesario para llamar a LEECAR

que “elimina” el carácter del buffer interno, se carga el carácter en TBA o TBB (dependiendo de la línea en la que se ha producido la interrupción), ya que en este registro se escribe el carácter que se va a enviar por la línea correspondiente. Si el buffer interno no está vacío se vuelve a la sección de identificación de la fuente de interrupción (ESP\_RTI).

Una vez que el buffer interno está vacío, se inhiben las interrupciones de transmisión de la línea correspondiente a través de la copia de IMR (IMR\_2) y se restauran los registros utilizados.

## CASOS DE PRUEBA

Se desarrollaron los primeros casos de prueba en torno al SCAN, ya que fue la primera subrutina desarrollada (después del INIT). El funcionamiento de todas estas era similar, introducir caracteres a un buffer y realizar una llamada a SCAN para intentar leer todos ellos correctamente. Realizamos también comprobaciones de error rutinarias, por ejemplo, que si se introdujesen parámetros incorrectos se terminase la subrutina.

Las pruebas pertenecientes al PRINT sucedieron después y de una manera similar, comprobando la gestión de errores y si el conteo devuelto por la subrutina era correcto.

Para apoyarnos en la resolución de casos de prueba desarrollamos unas subrutinas muy sencillas llamadas BIEN y MAL, que introducían distintos valores en el registro D0 al terminar la prueba.

# VALORACIONES FINALES

Para realizar este proyecto nos ha sido de gran utilidad el material de apoyo proporcionado: manual, grabaciones en zoom y el código auxiliar.

No ha resultado nada fácil la realización del proyecto, ya que para programar en ensamblador hay que ser muy minucioso con todas las instrucciones que estamos realizando.

Este proyecto nos ha ayudado a comprender a nivel fundamental las operaciones que realiza un ordenador.

El tiempo de realización de la práctica es complicado de estimar, ya que antes de ponernos a programar, dedicamos muchas horas a comprender el tema teórico de interrupciones, el lenguaje MC68k y la DUART MC68681.

Tiempo estimado:

- Preparación, estudio, presentaciones de zoom y comprensión del enunciado: 15h
- INIT: 3h
- SCAN y PRINT: 13h
- RTI: 5h
- Pruebas: 6h
  
- Tiempo total estimado: 42h

Para finalizar, queremos agradecer a los profesores por el apoyo que nos han dado a lo largo de la realización de la práctica y la rápida respuesta a los correos.