

Universidad Nacional Mayor de San Marcos
Facultad de Ingeniería de Sistemas e Informática

Escuela Profesional de Ingeniería de Software

Lima, Perú



Asignatura: Arquitectura de computadores

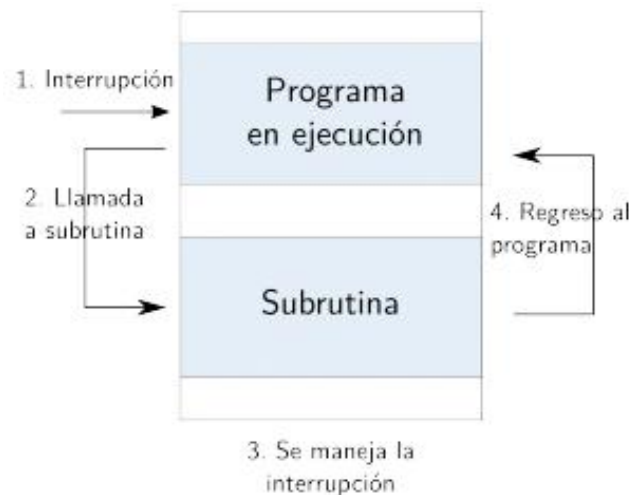
Interrupciones -
Entrada/salida de propósito general (GPIO)

Dr. Igor Aguilar Alonso

7. Interrupciones. Entrada/salida de propósito general - GPIO.

Interrupciones

- Los dispositivos de E/S requieren la atención del procesador.
- El procesador puede atender a los dispositivos por interrogación (Pooling) o por interrupciones.
- Las interrupciones son un **mecanismo para pausar brevemente** la ejecución de un **programa principal** que se está ejecutando, para ejecutar algún evento o una **subrutina** especial, y luego continuar ejecutando el programa principal.

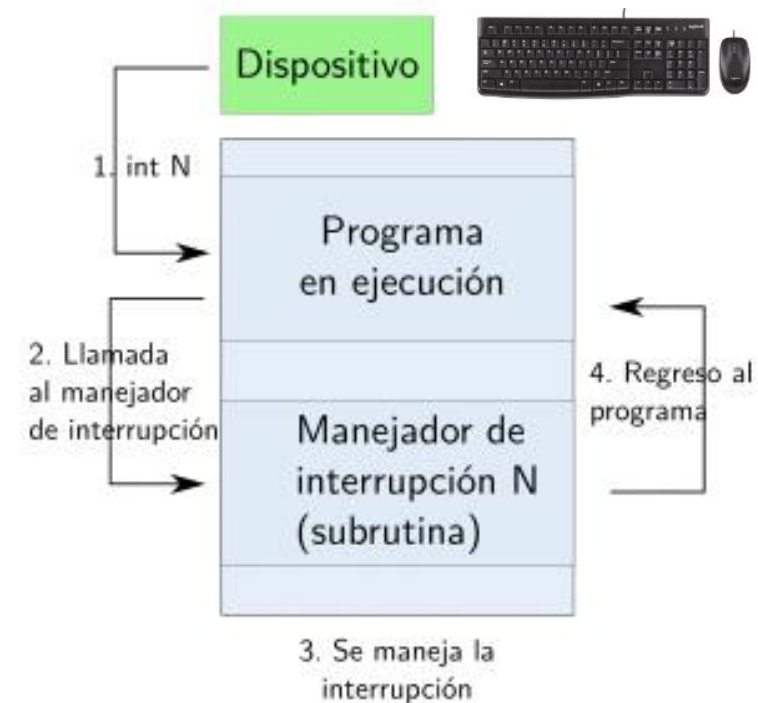


Hay dos tipos de interrupciones:

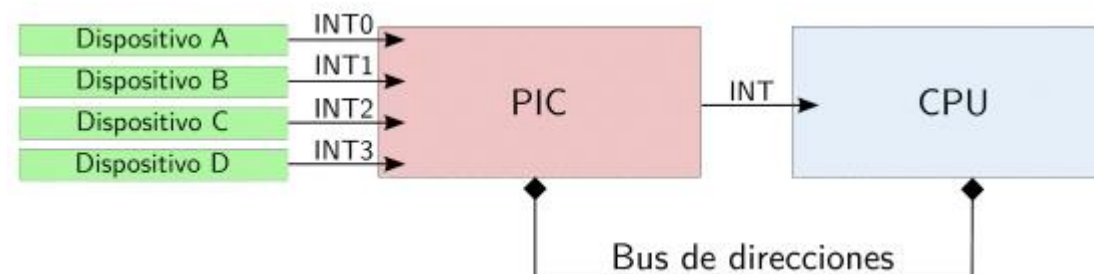
- *Interrupciones por hardware*
- *Interrupciones por software*

INTERRUPCIONES POR HARDWARE

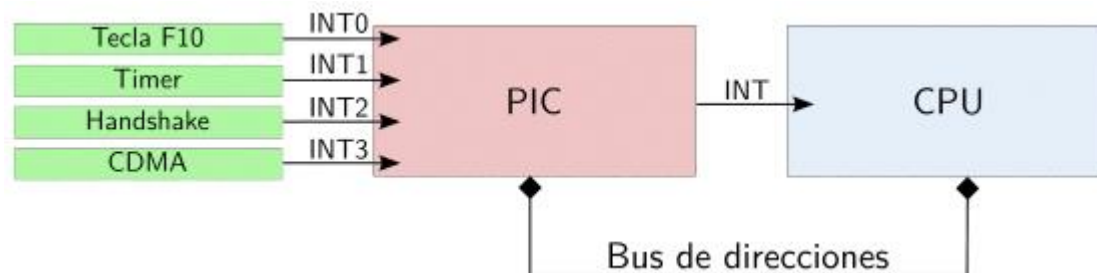
- Sirven para atender eventos de los dispositivos de E/S.
- Los dispara el dispositivo, y causan que se ejecute una subrutina del programa (debe implementarla el programador).
- Las interrupciones se *identifican por un número*, entre 0 y 255 (256 definidas), llamado identificador de interrupción.
- El mecanismo de interrupciones debe permitir hacer entonces una asociación entre los *identificadores de interrupción* y las subrutinas (en realidad, entre los *identificadores* y las *direcciones* de comienzo de las subrutinas).



- El **Controlador de Interrupciones Programable** (PIC), es un dispositivo intermediario entre la CPU y las interrupciones de los dispositivos, que de alguna manera permita conectar los ‘cables’ de varios dispositivos a la CPU de forma indirecta.
- El PIC, es el periférico encargado de administrar los pedidos de interrupción procedentes de los demás periféricos.



- Conexiones de dispositivos (F10, Timer, Handshake, CDMA) al PIC en el MSX88



Reloj Interno:

Un ejemplo claro de este tipo de interrupción es la que actualiza el **contador del reloj interno** de la computadora.

El hardware hace el llamado a esta interrupción varias veces durante un segundo para mantener la hora actualizada¹.

Teclado y Mouse:

Las pulsaciones de un **teclado** o la acción de un **mouse** invocan interrupciones de hardware.

Cuando presionamos una tecla, el teclado envía una señal al procesador para procesarla.

De manera similar, cuando movemos el mouse, se generan interrupciones para manejar esos eventos asincrónicos².

Dispositivos de E/S:

- Los dispositivos de entrada/salida (E/S) como **discos duros, tarjetas de red o puertos USB** generan interrupciones.
- Por ejemplo, cuando se completa una operación de lectura o escritura en un disco, se activa una interrupción para notificar al procesador.

Controladores de Periféricos:

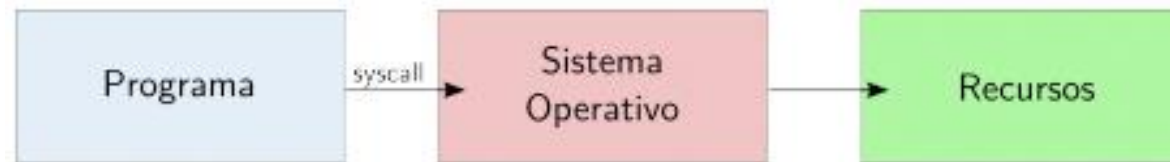
- Los **controladores de hardware** (drivers) también utilizan interrupciones.
- Por ejemplo, un controlador de impresora puede generar una interrupción cuando la impresión se ha completado o cuando hay un atasco de papel.

Comunicación Serial:

- Al recibir datos a través de **puertos seriales** (como RS-232 o USB), se generan interrupciones para procesar esos datos.
- Esto es común en aplicaciones como la comunicación con módems o dispositivos GPS

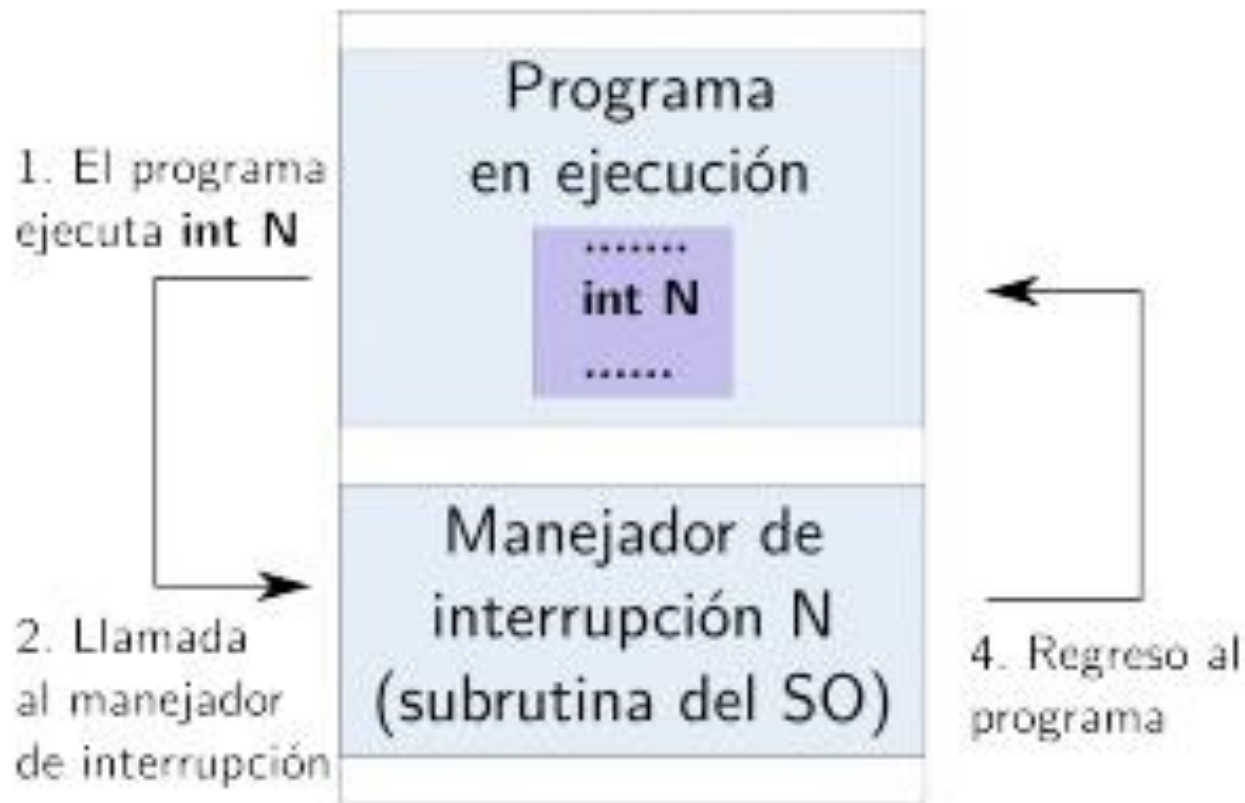
INTERRUPCIONES POR SOFTWARE

- Los programas hacen *llamadas al sistema operativo* (*syscall*).
- Se interrumpe la ejecución del programa y se pasa temporariamente el control a una subrutina del SO. El SO satisface el pedido del programa (*ejecutando las instrucciones de la subrutina del SO*), y luego continúa con la ejecución del mismo.



- El mecanismo para realizar una *syscall* (para la ejecución de una subrutina del SO) es muy similar a llamar a una subrutina o procedimiento; debemos conocer la dirección de memoria donde se encuentra la misma, y poner en el registro IP dicha dirección (y apilar la dirección de retorno, para que cuando termine la subrutina del SO se pueda retornar la ejecución al punto correcto del programa!).

- Las inicia el programa mediante la instrucción **int N**, donde N es el numero de interrupciones, y hacen que se ejecute una subrutina del sistema operativo (cuyo código está oculto al programador).



Las **interrupciones por software** se pueden clasificar en diferentes categorías según su origen y función. A continuación, te presento algunos tipos preasignados:

1. **Llamadas al Sistema:**

Las **llamadas al sistema** son interrupciones de software utilizadas para solicitar servicios del sistema operativo. Ejemplos de llamadas al sistema incluyen:

- Abrir y cerrar archivos.
- Crear y eliminar procesos.
- Enviar y recibir datos.

2. **Excepciones:**

- Las excepciones también son interrupciones de software.
- Por ejemplo, cuando ocurre una **división por cero** o se accede a una **dirección de memoria no válida**, se genera una excepción.
- El sistema operativo maneja estas excepciones para evitar que el programa se bloquee

- Hay 4 tipos de interrupción por software preasignadas:
 - ***Interrupción 0 (HLT): Terminar el programa.***
 - ***Interrupción 3 (debug): Poner un punto de parada (breakpoint).***
 - ***Interrupción 6 (leer caracter): Lee un carácter de teclado.***
 - ***Interrupción 7 (imprimir un string): Escribe un string en pantalla***

*En resumen, las interrupciones por software son esenciales para la **comunicación entre programas** y **el sistema operativo**, permitiendo que los programas accedan a servicios específicos y manejen situaciones excepcionales.*

Introducción a las entradas y salidas de propósito general (GPIO)

Todos los sistemas incluyen y utilizan dos grupos de dispositivos más comúnmente usados:

- la entrada/salida de propósito general y
- Los dispositivos de gestión del tiempo.

Las entradas y salidas de propósito general (**General Purpose Input Output - GPIO**) es un sistema que consta de una serie de **pines** o **conexiones** que se pueden usar como entradas o salidas *para múltiples usos con el microprocesador*.

Todos los sistemas incluyen y utilizan estos tipos de dispositivos, que son fundamentales en los **sistemas empuetrados** y en los **microcontroladores**.

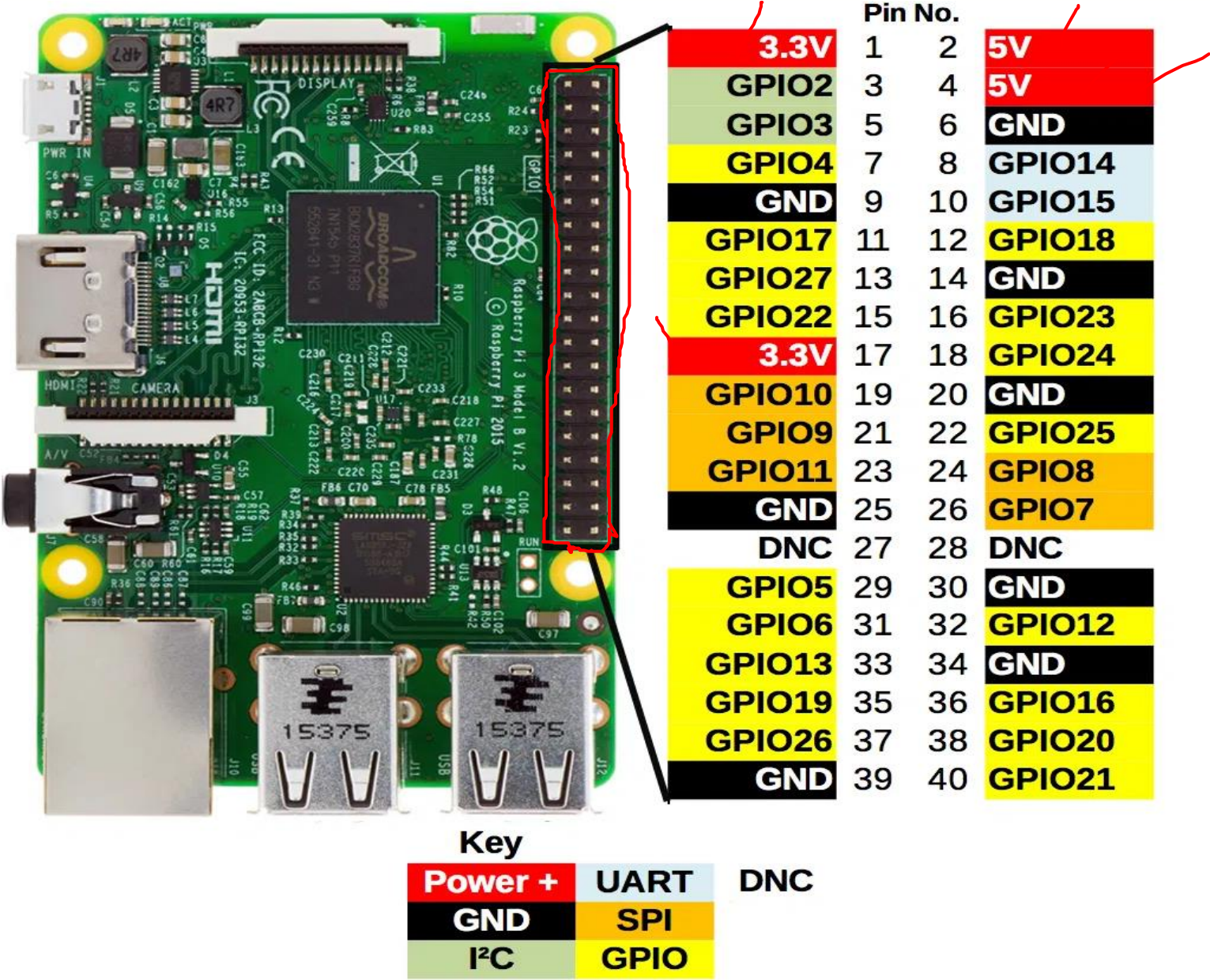



Figura de un Raspberry PI -
GPIOs - MBrobotics

Las entrada/salida de propósito general

- Las entradas/salidas de propósito general conocida por los términos en ingles - *General Purpose Input Output (GPIO)* .
- Es la forma más sencilla de entrada/salida que podemos encontrar en un **procesador con sus propios pines de conexión** eléctrica con el exterior (con otros dispositivos).
- Está presente en todos los sistemas informáticos.
- Los **PC actuales** la utilizan para leer pulsadores o encender algún LED del chasis.
- En los **microcontroladores**, que son sistemas completos en un chip, *la GPIO tiene más importancia* y muestra su mayor complejidad y potencia.

- Si la organización del procesador permite relacionar direcciones del mapa de memoria o de entrada salida con algunos pines, la escritura de un **1** o **0** lógicos por parte de un programa - arquitectura en esas direcciones se reflejará en cierta tensión eléctrica en el pin, normalmente *0V para el nivel bajo* y *5 o 3,3V para el nivel alto*, que puede ser utilizada para activar o desactivar algún dispositivo externo.
- **Por ejemplo**, esto nos permitiría encender o apagar un LED mediante instrucciones de nuestro programa.
- De modo análogo, en el caso de *las entradas*, si el valor eléctrico presente en el pin se ve traducido por el diseño eléctrico del circuito en un *1 o 0 lógico* que se puede leer en una dirección del sistema, podremos detectar cambios en el exterior de nuestro procesador.
- De esta manera, por ejemplo, *un programa* podrá *consultar si un pulsador está libre u oprimido*, y tomar decisiones en función de su estado.

Veamos en este sencillo ejemplo:

09_entrada_salida.s 

```
1      ldr    r0, [r7, #PULSADOR]    @ Leemos el nivel
2      cmp    r0, #1                 @ Si no está pulsado
3      bne    sigue                  @ seguimos
4      mov    r0, #1                  @ Escribimos 1 para
5      str    r0, [r7, #LED]          @ encender el LED
```

- En el fragmento de código anterior supuestamente enciende un LED **escribiendo un 1** en la **dirección «r7 + LED»** si el pulsador está presionado, es decir, cuando lee un 1 en la dirección «r7 + PULSADOR».
- Es un ejemplo figurado que simplifica el caso real.

La GPIO en la E/S de los sistemas

- La GPIO (General Purpose Input Output) es tan útil y necesaria que **está presente en todos los sistemas informáticos**.
- Los **PC actuales** la utilizan para leer pulsadores o encender algún LED del chasis.
- Por otra parte, en los **microcontroladores** (**sistemas completos en un chip**), la GPIO tiene más importancia y muestra su mayor complejidad y potencia.

A continuación vamos a analizar los aspectos e implicaciones de la GPIO y su uso en estos sistemas.

Aspectos lógicos y físicos de la GPIO o programación y electrónica de la GPIO

- En la mayor parte de los sistemas, diversos pines de entrada/salida se agrupan en **palabras**, de tal forma que cada acceso como los del ejemplo afectaría a todos los pines asociados a la palabra a cuya dirección se accede.
- De esta manera, se habla de **puertos** refiriéndose a cada una de las **direcciones asociadas** a *conjuntos de pines en el exterior del circuito*, y cada pin individual es un bit del puerto.
- Así, por ejemplo, si hablamos de **PB12** en el **microcontrolador ATSAM3X8E**, nos estamos refiriendo al **bit 12** del puerto de salida llamado PB, que físicamente se corresponde con el pin 86 del encapsulado LQFP del microcontrolador, algo que es necesario saber para diseñar el hardware del sistema.

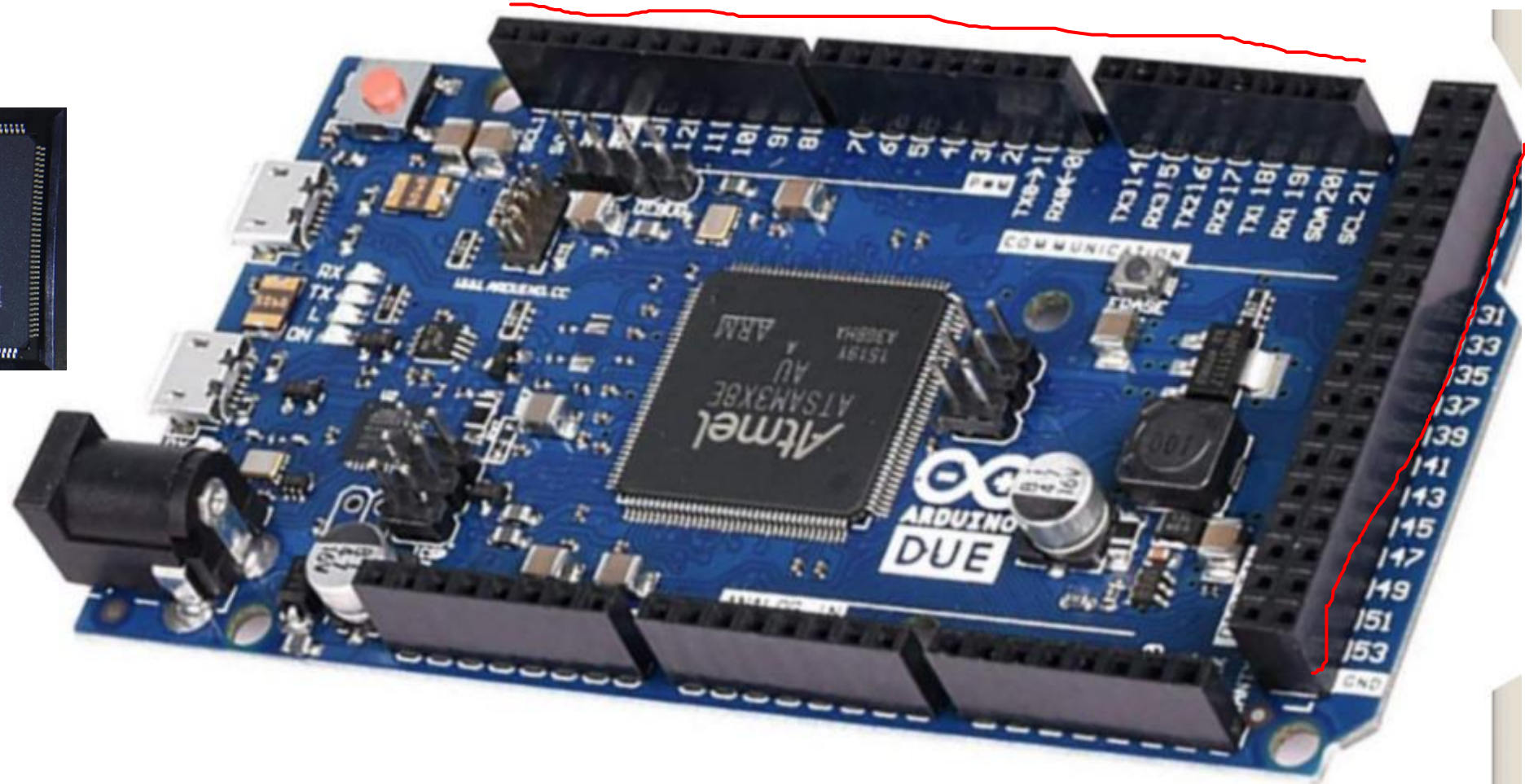
- En este caso, para actuar modificar o comprobar su valor sobre bits individuales o sobre conjuntos de bits es necesario utilizar **máscaras** y **operaciones lógicas** para no afectar a otros pines del mismo puerto.
- Suponiendo que el LED se encuentra en el **bit 12** y el pulsador en el bit 20 del citado puerto PB, una versión más verosímil del ejemplo propuesto sería:

```
09_acceso_es.s
1      ldr    r7, =PB          @ Dirección del puerto
2      ldr    r6, =0x00100000 @ Máscara para el bit 20
3      ldr    r0, [r7]         @ Leemos el puerto
4      ands   r0, r6           @ y verificamos el bit
5      beq    sigue           @ Seguimos si está a 0
6      ldr    r6, =0x00001000 @ Máscara para el bit 12
7      ldr    r0, [r7]         @ Leemos el puerto
8      orr     r0, r6          @ ponemos a 1 el bit
9      str    r0, [r7]         @ y lo escribimos en el puerto
```

- En este caso, en primer lugar se accede a la dirección del puerto PB para leer el estado de todos los bits y, mediante una máscara y la operación lógica AND, se verifica si el bit correspondiente al pulsador bit 20 está a 1.
- En caso afirmativo, cuando el resultado de AND no es cero se lee de nuevo PB y mediante una operación OR y la máscara correspondiente se pone a 1 el bit 12, correspondiente al LED para encenderlo.
- La **operación OR** permite, en este caso, poner a **1 un bit sin modificar** los demás.

Este ejemplo es más cercano a la realidad y sería válido para muchos microcontroladores, el caso del **ATSAM3X8E** que es algo más complejo.

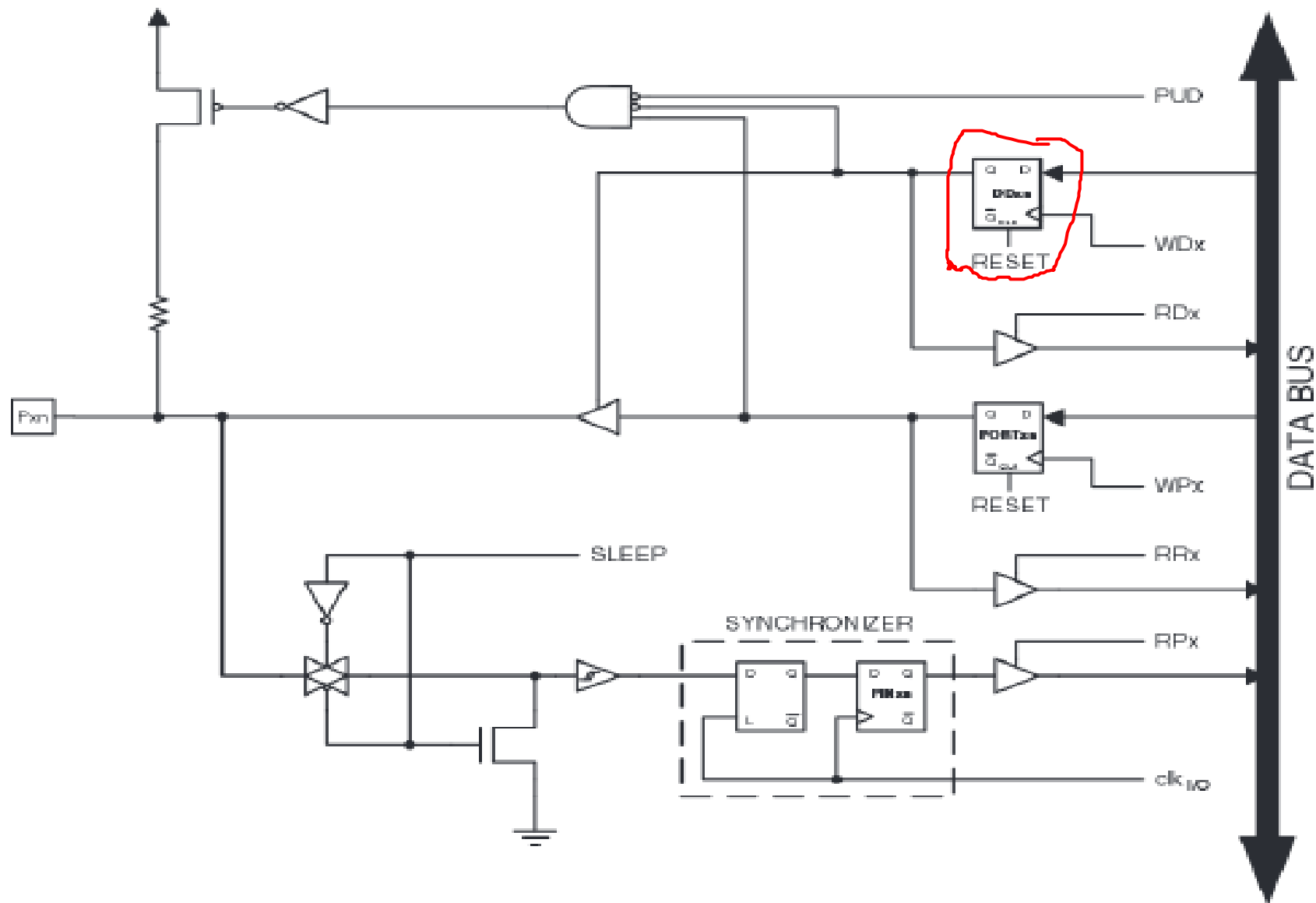
Microcontrolador ATSAM3X8E.



- El ejemplo que se acaba de presentar es válido para mostrar la gestión por programa de la entrada y salida tipo GPIO.

- No olvidemos que los pines se relacionan con el exterior mediante magnitudes eléctricas, el un pin que funciona como **entrada** es totalmente distinto al que se utiliza como **salida**, previamente hace falta una configuración de los puertos de la GPIO en que se indique qué pines van a actuar como entrada y cuáles como salida.
- Así, asociado a la dirección en la que se **leen o escriben** los datos y que hemos llamado **PB** en el ejemplo, habrá al menos otra que corresponda a un **registro de control de la GPIO** en la que se indique qué pines se comportan como entradas y cuáles como salidas, lo que se conoce como **dirección de los pines**.
- La GPIO a veces utiliza dos puertos, con direcciones distintas, para leer o escribir en los pines del sistema.
- El **registro o latch**, que se asocia a las salidas suele tener una **dirección** y las **entradas** que no requieren registro pues leen el valor lógico fijado externamente en el pin.

- En el caso más común, un puerto GPIO ocupa al menos **tres direcciones en el mapa:** una para el **registro de control** que configura la dirección de los pines, otra para el **registro de datos de salida** y otra para leer directamente los pines a través del **registro de datos de entrada**.
- En la siguiente Figura (obtenida del manual [Atm11]) se muestra la estructura interna de un pin de E/S de un microcontrolador de la familia Atmel AVR.



Estructura interna de un pin de E/S de un microcontrolador de la familia Atmel AVR

Interrupciones asociadas a la GPIO

- Las interrupciones son una forma de **sincronizar** el **procesador** con los **dispositivos de entrada/salida** para que éstos puedan avisar de forma asíncrona al procesador de **que requieren su atención**, sin necesidad de que aquél se preocupe periódicamente de atenderlos, lo que sería encuesta o prueba de estado.
- Los *sistemas avanzados de GPIO* incorporan la posibilidad de avisar al procesador de ciertos cambios mediante interrupciones, para poder realizar su gestión de forma más eficaz.
- Existen dos tipos de interrupciones que se pueden asociar a la GPIO, por supuesto siempre utilizada como entrada, como veremos a continuación.

- *En primer lugar*; se pueden utilizar los pines como líneas de interrupción, bien para señalar un cambio relativo al circuito conectado al pin, como oprimir un pulsador, bien para conectar una señal de un circuito externo y que así el circuito sea capaz de generar interrupciones.
- En este último caso el pin de la GPIO haría el papel de una línea de interrupción externa de un procesador.
- En ambos casos suele poder configurarse si la interrupción se señala por nivel o por flanco y su polaridad.
- *En segundo lugar*, y asociado a las características de bajo consumo de los microcontroladores, se tiene la interrupción por cambio de valor.
- Esta interrupción puede estar asociada a un pin o un conjunto de ellos, y se activa cada vez que alguno de los pines de entrada del grupo cambia su valor, desde la última vez que se leyó.

- Esta interrupción, además, **suele usarse para sacar al procesador de un modo de bajo consumo y activarlo otra vez para reaccionar frente al cambio indicado.**
- El uso de interrupciones asociadas a la GPIO requiere añadir nuevos registros de control y estado, para configurar las interrupciones y sus características – control- y para almacenar los indicadores – *flags* - que informen sobre las circunstancias de la interrupción.

Aspectos avanzados de la GPIO

Además de las interrupciones y la relación con los modos de bajo consumo, los microcontroladores avanzados añaden características y complejidad, a sus bloques de GPIO, estas características dependen bastante de la familia de microcontroladores, algunas tendencias generales que se comentan a continuación.

- En primer lugar tenemos las modificaciones eléctricas de los bloques asociados a los pines. Estas modificaciones afectan solo a subconjuntos de estos pines y en algunos **casos no son configurables**, por lo que se deben tener en cuenta fundamentalmente en el diseño electrónico del sistema.
- También están pines de entrada que soportan varios umbrales lógicos, lo más normal es **5V** y **3,3V** para el nivel alto.

- Es frecuente encontrar entradas con disparador de Schmitt para generar flancos más rápidos en las señales eléctricas en el interior del circuito, por ejemplo en entradas que generen interrupciones, lo que produce que los valores VIHMIN y VILMAX en estos pines estén más próximos, reduciendo el rango de tensiones indeterminadas a nivel lógico entre ellos.
- Tenemos salidas que pueden configurarse como **colector abierto**, (open drain), lo que permite utilizarlas en sistemas AND cableados, muy utilizados en buses.
- Otra tendencia actual, es utilizar un muestreo periódico de los pines de entrada, que requiere almacenar su valor en un registro, en lugar de valor presente en el pin en el la lectura.
- Es posible **añadir filtros** que permitan tratar ruido eléctrico en las entradas o eliminar los rebotes típicos en los pulsadores e interruptores. se incorporan a la GPIO registros para activar o configurar estos métodos de filtrado.

- La forma de tratar las entradas **requiere un reloj** para muestrearlas y almacenar su valor en el registro, lo que requiere parar este reloj para reducir el consumo eléctrico.
- La última característica asociada a la GPIO surge de la necesidad de versatilidad de los microcontroladores. Los dispositivos actuales, con **gran número de pines** en su GPIO, incorporan **convertidores ADC y DAC**, **buses** e **interfaces estándar**, etc., que necesitan de pines específicos para relacionarse con el exterior.
- Todo esto deja en libertad al diseñador de seleccionar la configuración del sistema adecuada para su aplicación, muchos pines pueden usarse como parte de la GPIO o con alguna de estas funciones específicas. Esto hace un complejo subsistema de encaminado de señales entre los dispositivos internos y los pines, que afecta directamente a la GPIO.

¿Preguntas ...?

Gracias por su atención

iaquilara@unmsm.edu.pe

[Igor Aguilar Alonso](#)

Análisis de lectura

Analizar la lectura de interrupciones en grupo:

- Preparar un informe
- Exponer en clase