

Gestione di Input/Output

Un sistema di **Input/Output** permette di comunicare con il mondo esterno.

Ci sono due modi per gestire questo tipo di richieste:

- **polling**
- **interrupt**

Eventi di sistema

Ci sono diverse categorie:

- **Infrequenti ma importanti:** Come le **NMI** o **Hard Fault**
- **I/O Synchronization:** Triggera un interrupt quando un segnale su una porta cambia E.g. premendo un pulsante
- **Interrupt periodici:** Generati da timer a intervalli regolari
- **Conversione ADC:** Da analogico a digitale o viceversa

Polling

E' un **processo** dove il computer o un device apposito aspetta che un device esterno controlli dei **registri di stato**.

Di solito è implementato come un ciclo infinito dove il *controlling device* controlla in continuazione dei registri appositi.

A causa di questo **loop**, il **polling** non è proprio il massimo a livello di efficienza(per un sistema **bare metal**) e ha una bassa performance per quanto riguarda la gestione di *richieste annidate*.

Interrupt

Le periferiche comunicano **direttamente** con la **CPU**.

La **CPU** entra in **idle mode** quando non sta gestendo le richieste, e si sveglia appena una periferica richiede un servizio.

Si basa sulla **IVT** e si appoggia ad un device esterno chiamato **Interrupt Controller**

Setup per interrupt mode

Ci sono diverse cose da fare per *setappare* un sistema per farlo lavorare in *interrupt mode*.

Boot Time

Bisogna:

- Inizializzare **strutture dati** quali **contatori** etc
- Configurare l'**interrupt controller**

Run Time

In ogni routine di gestione di interrupt bisogna:

- *Clearare* i flag che indicano che l'interrupt è attivo(ogni tanto viene fatto in automatico, dipende dalle periferiche)
- Mantenere i registri **R4-R8,R10-R11(ABI AAPCS)**
- Comunicare attraverso **variabili globali condivise**

Interrupt Controller

E' un device che combina diverse sorgenti di interrupt in una o più **CPU Lines**.

Gestisce i segnali di interrupt ricevuti dai diversi devices combinandoli in un singolo **interrupt output**.

Gestisce anche le priorità delle varie interrupt.

Nested Vectored Interrupt Controller (NVIC)

E' una parte integrante del **Cortex-M3**. Il suo alto **coupling** con la **CPU** permette una gestione delle **interrupt** a bassa latenza.

Gestisce fino a 35 **interrupt** esterne.