

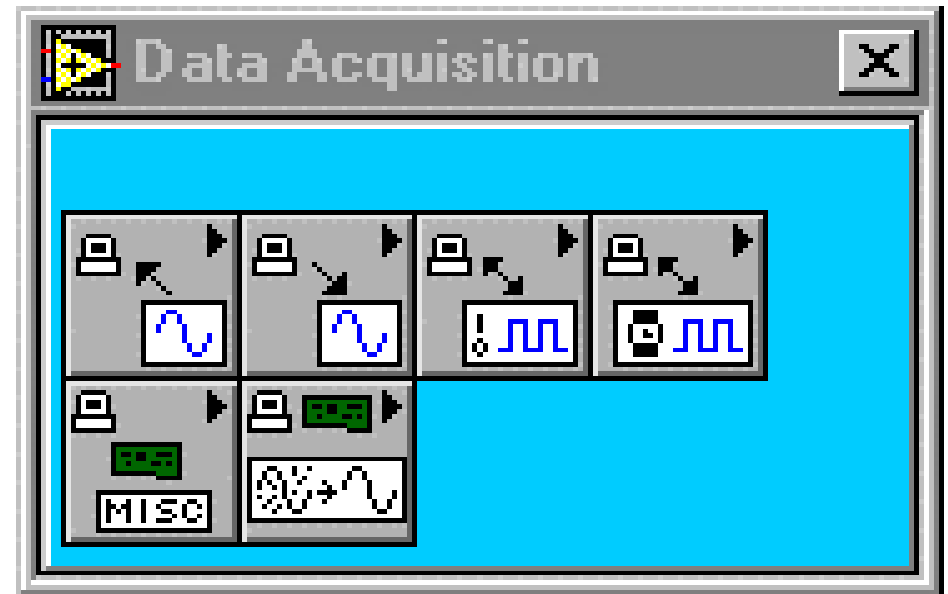
# *Corso di Sistemi Automatici di Misura*

**QUARTA LEZIONE:**

**L'ACQUISIZIONE DATI IN  
LABVIEW**

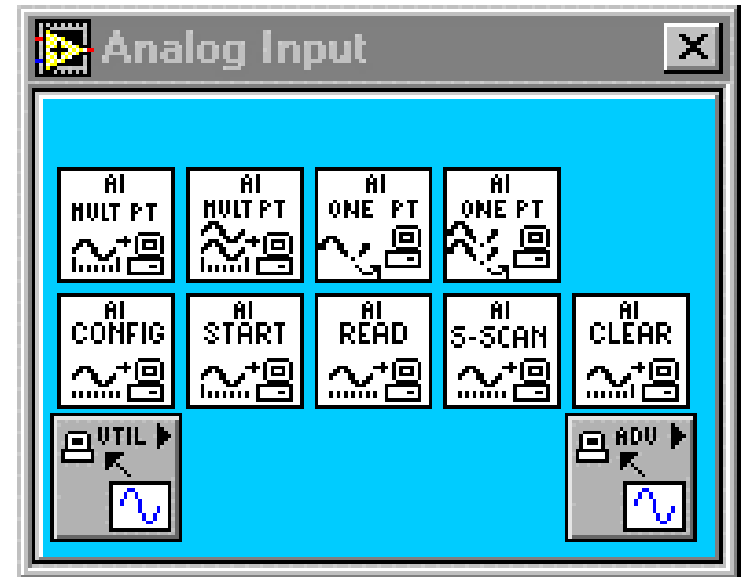
# Organizzazione delle funzioni LabView

- Analog Input
- Analog Output
- Digital I/O
- Contatori
- Calibrazione e configurazione
- Condizionamento dei segnali

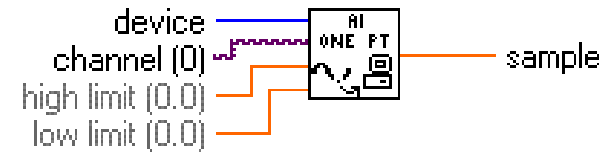
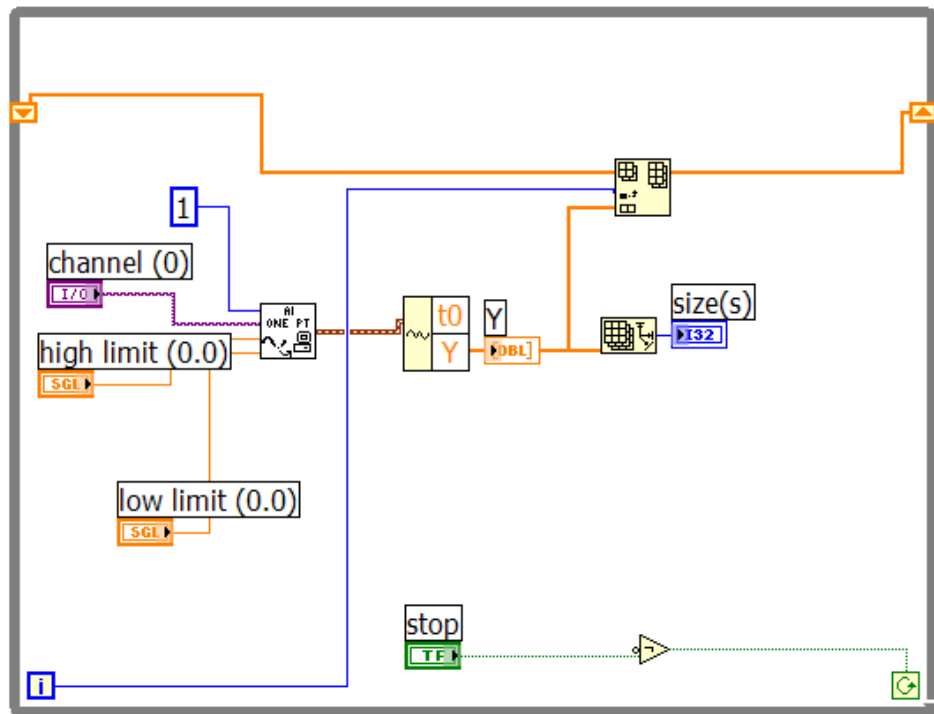


# VI per gli ingressi analogici

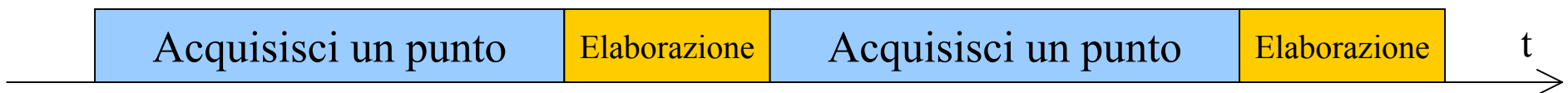
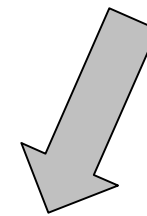
- **VI di alto livello**
  - Gestione di una acquisizione tipica
  - Semplici da usare,
  - Non consentono un controllo di ogni funzione della scheda
- **VI di livello intermedio**
  - Controllo di tutte le funzioni di acquisizione
  - Acquisizione continua, anche con *trigger*
  - Scrittura dei dati acquisiti su disco (*streaming*)



# Acquisizione continua con funzioni di alto livello (VI Easy)

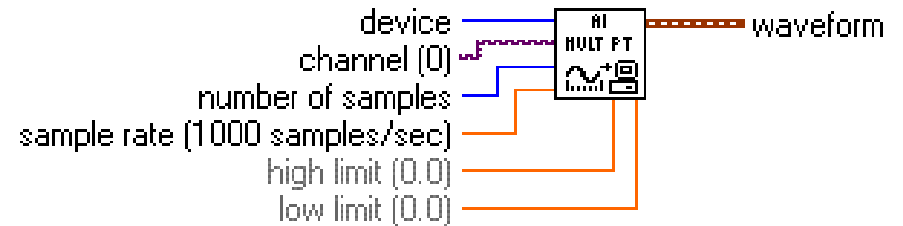
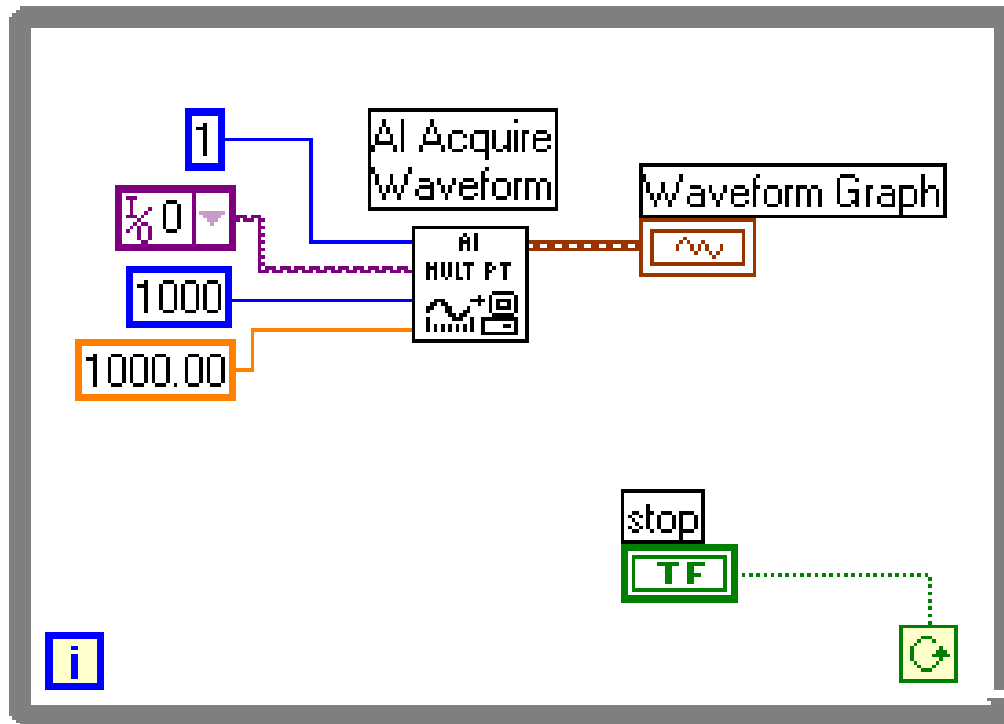


L' acquisizione non è effettuata durante l'elaborazione

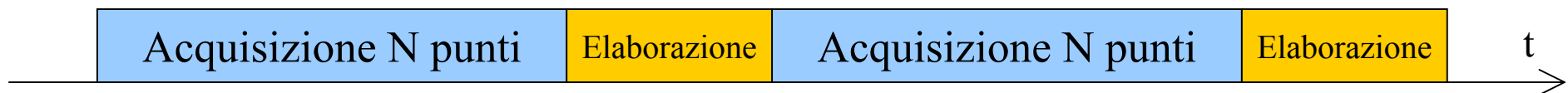
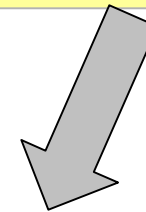


Il periodo di campionamento non è controllabile dall'utente!

# Acquisizione continua con funzioni di alto livello (VI Easy)



La acquisizione non è effettuata durante l'elaborazione

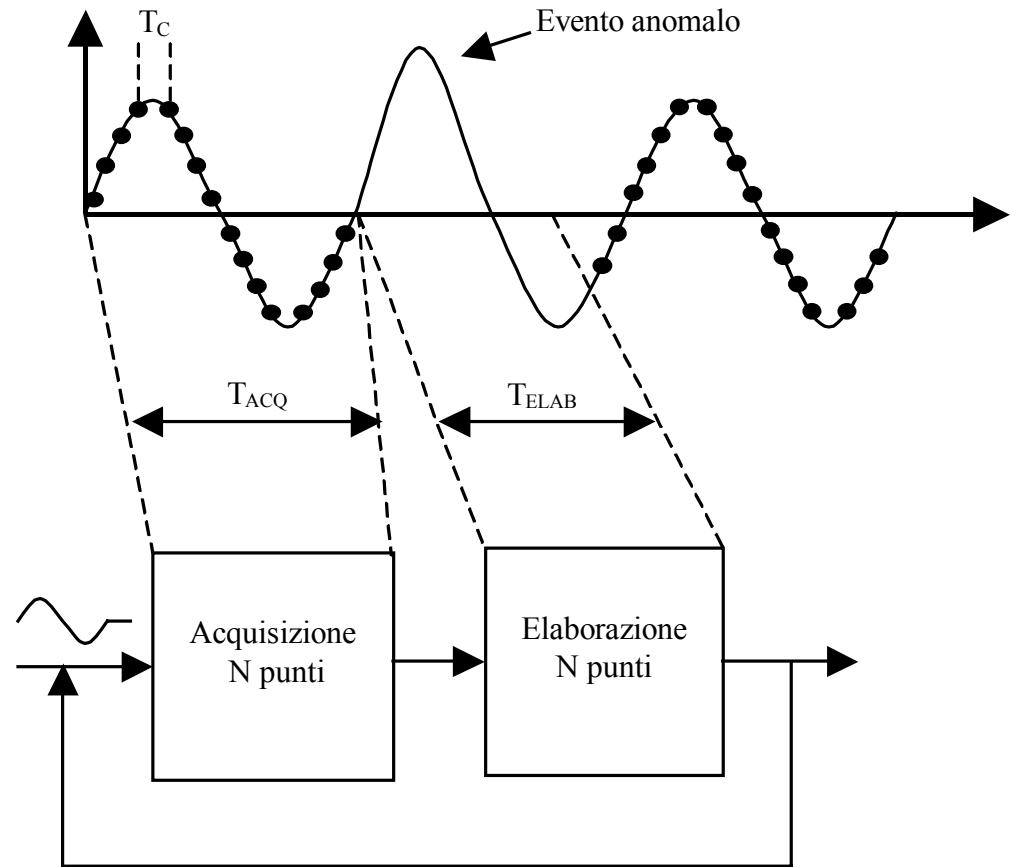
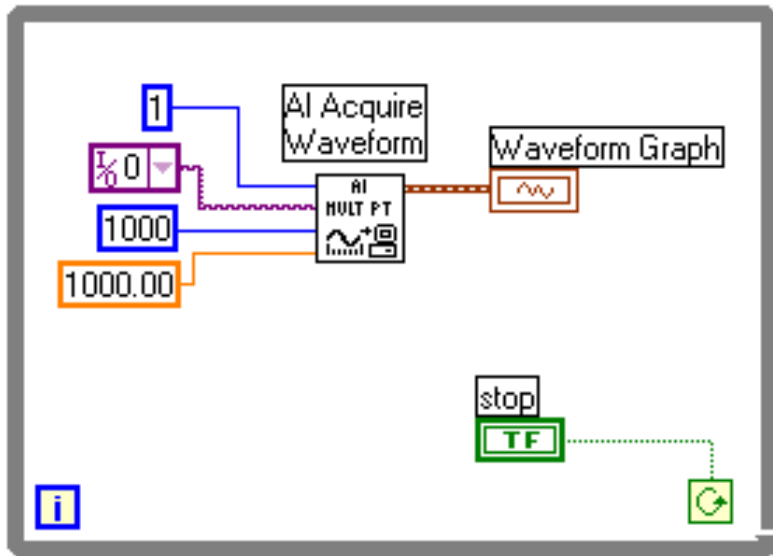


I campioni vengono memorizzati all'interno di un buffer di memoria (allocato nella memoria RAM del PC) di dimensione variabile dall'utente

E' possibile scegliere il periodo di campionamento

L'elaborazione ritarda l'acquisizione dei prossimi N punti

# Acquisizione continua con funzioni di alto livello (VI Easy)

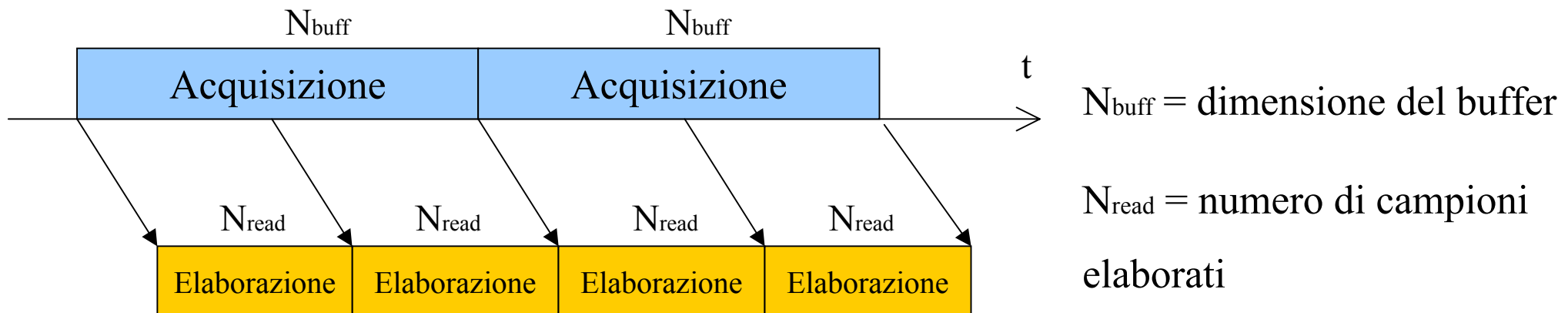


Potrebbero essere persi degli eventi importanti!!

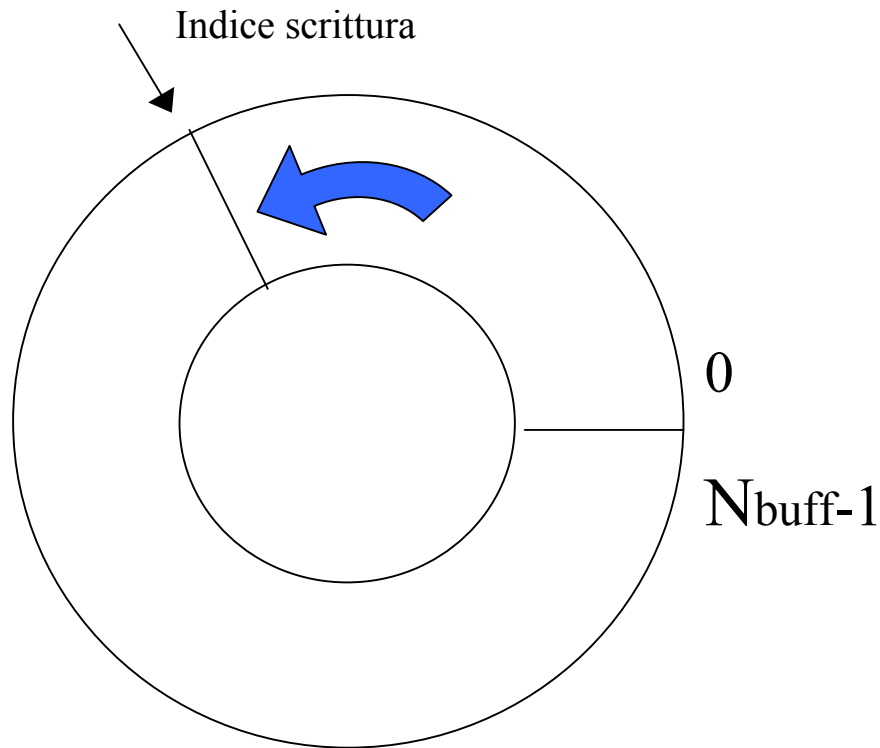
# Acquisizione continua con buffer circolare

Per evitare di perdere dei campioni, il buffer di acquisizione (allocato nella memoria RAM del PC) può essere opportunamente gestito in modo da:

- 1) Fare un uso efficiente della memoria
- 2) Consentire l'elaborazione di un set di campioni contemporaneamente all'acquisizione di campioni successivi senza interrompere il processo di acquisizione



# Acquisizione continua con buffer circolare



## SCRITTURA

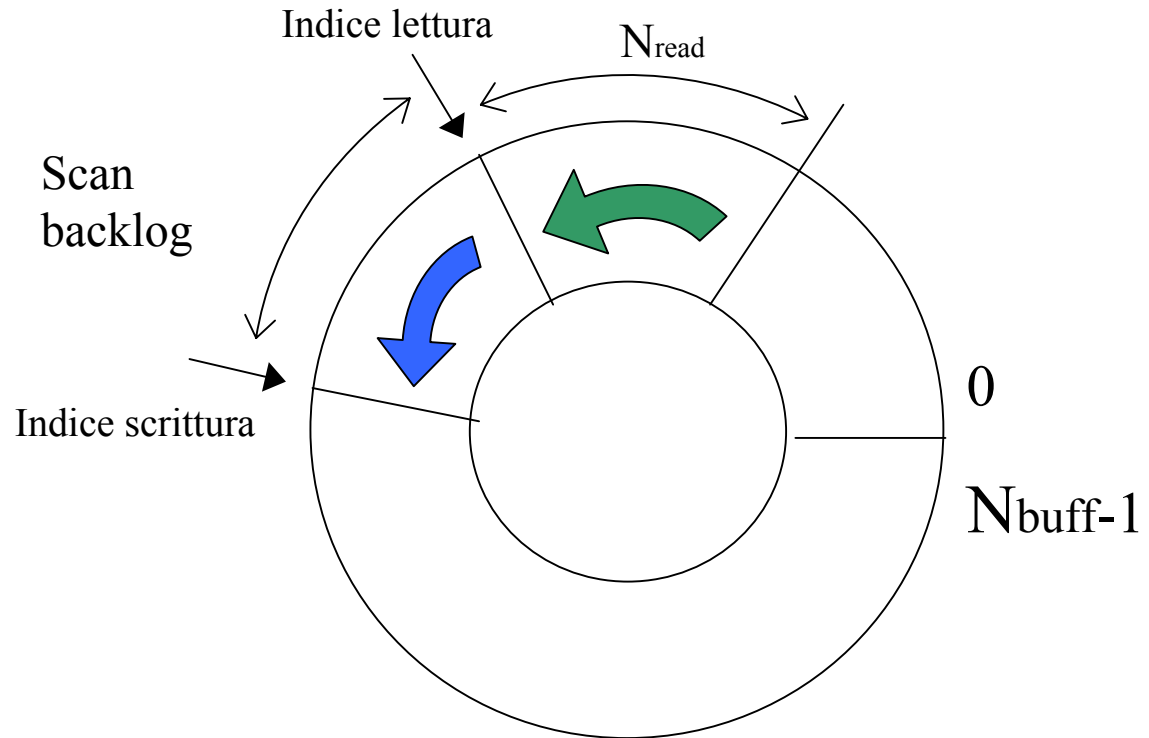
- ✓ E' riempito dall'*hardware* di acquisizione
- ✓ Raggiunta la fine del *buffer*, la scrittura riprende dalla posizione 0, con sovrascrittura dei dati precedentemente acquisiti



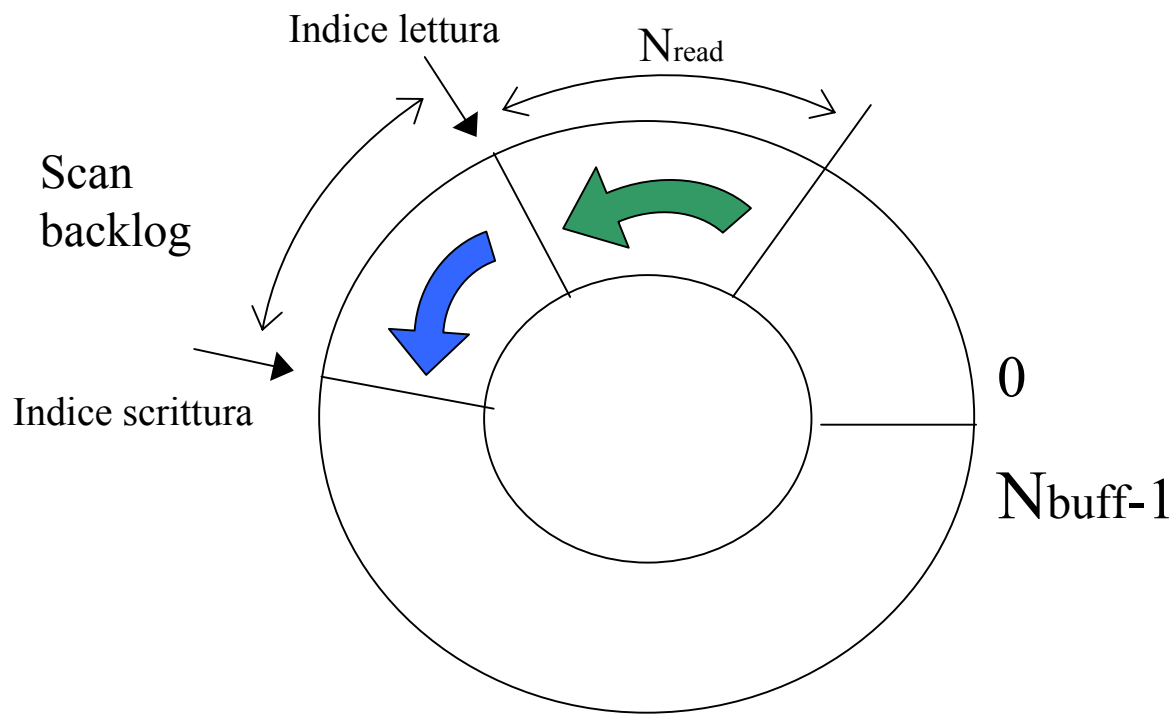
# Acquisizione continua con buffer circolare

## LETTURA

- ✓ Il *software* legge i dati ( $N_{\text{read}}$  punti alla volta) in modo contemporaneo alla acquisizione
- ✓ La *velocità di lettura* dipende dal tempo necessario alla elaborazione di ogni sequenza di  $N_{\text{read}}$  punti
- ✓ La *velocità di lettura* deve essere uguale o maggiore a quella di scrittura altrimenti ci sarà sovrascrittura dei dati



# Acquisizione continua con buffer circolare

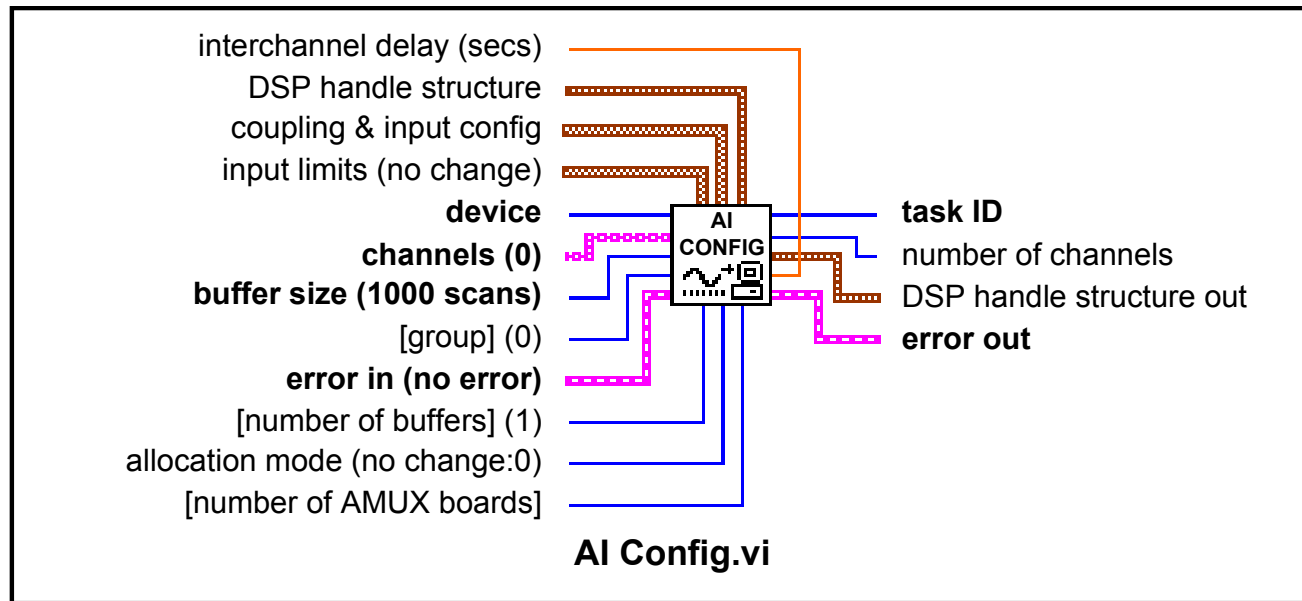


Si deve monitorare lo *Scan backlog* che rappresenta la distanza (in termini di campioni) tra l'indice di lettura e quello di scrittura. Se tale distanza supera  $N_{buff}$  (la velocità di lettura del buffer è molto minore di quella di scrittura) vengono sovrascritti dati non ancora letti.

Pertanto, i parametri dell'acquisizione:  $f_c$ ,  $N_{buff}$ ,  $N_{read}$  devono essere regolati in modo da mantenere lo *scan backlog* sufficientemente minore di  $N_{buff}$ .

Se possibile bisognerebbe contenere il tempo di elaborazione curando l'efficienza del codice.

# Acquisizione continua con funzioni di medio livello (VI Intermediate)



**AI CONFIG:** configura l'operazione di input analogico per un set di canali definito ed alloca un buffer nella memoria del computer.

**Principali parametri in ingresso:**

**channels:** è una stringa di caratteri che serve a definire i canali che verranno utilizzati per l'acquisizione;

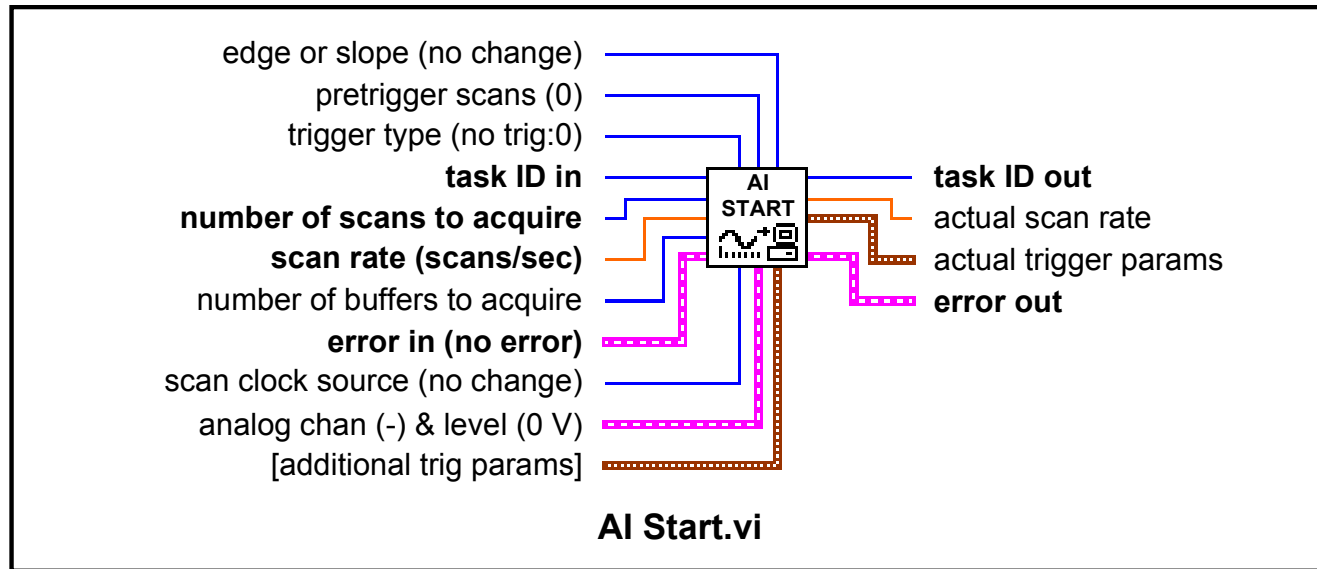
**buffer size:** è un numero intero che definisce la dimensione del buffer in base al numero di scansioni che si intende effettuare ed alloca la memoria necessaria.

**Principali parametri in uscita:**

**task ID:** è un numero intero, associato al device ed ai canali e serve ad identificarli;

**Error Out:** è un cluster contenente tutte le informazioni sugli errori

# Acquisizione continua con funzioni di medio livello (VI Intermediate)



**AI START:** inizia l'acquisizione bufferizzata

**Principali parametri in ingresso:**

**task ID:** è il numero intero definito in precedenza; poiché esso è un input ed un output per i vari VI che seguono AI Config.VI, si crea una dipendenza sequenziale tra i vari VI che effettuano l'acquisizione garantendo in questo modo una corretta sequenzialità nelle varie fasi del processo di acquisizione;

**number of scans to acquire:** è un numero intero che definisce il numero di scansioni che si intende effettuare per ogni canale. Se si impone che questo parametro valga 0, LabVIEW acquisisce i dati e li trasferisce nel buffer senza interruzione;

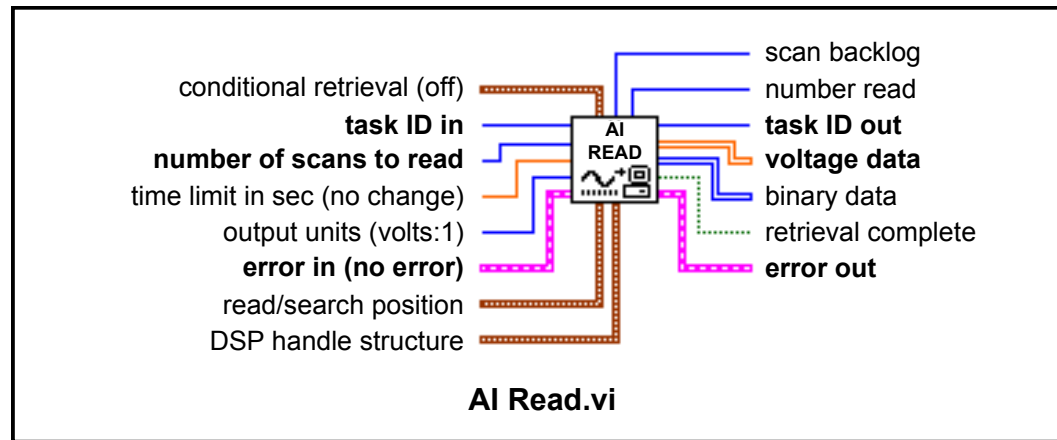
**scan rate:** è un numero che definisce la frequenza di acquisizione per ogni canale.

**Principali parametri in uscita:**

**task ID:** è un numero intero, associato al device ed ai canali e serve ad identificarli;

**Error Out:** è un cluster contenente tutte le informazioni sugli errori

# Acquisizione continua con funzioni di medio livello (VI Intermediate)



**AI READ:** legge i dati dal buffer allocato da Ai Config.VI

**Principali parametri in ingresso:**

**task ID:** (definito in precedenza);

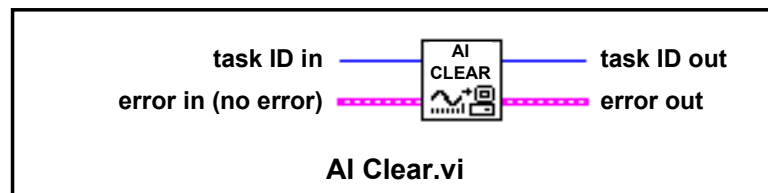
**number of scans to read:** è un numero intero che definisce il numero di punti che si intende leggere dal buffer.

**Principali parametri in uscita:**

**task ID** (definito in precedenza);

**error Out** (definito in precedenza);

**scaled Data** è un matrice bidimensionale che contiene i dati letti dal buffer (ogni colonna di dati è associata al corrispondente canale indicato nella Channel list).



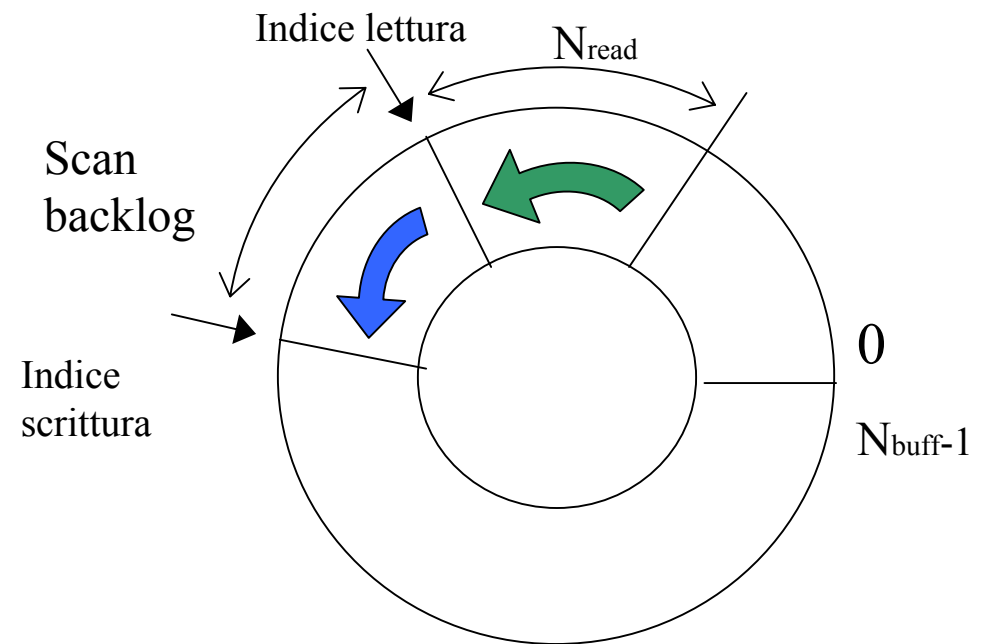
**AI CLEAR:** cancella il buffer e disalloca tutte le risorse coinvolte nell'acquisizione.

# Acquisizione continua con funzioni di medio livello (VI Intermediate)

## ...Riassumendo

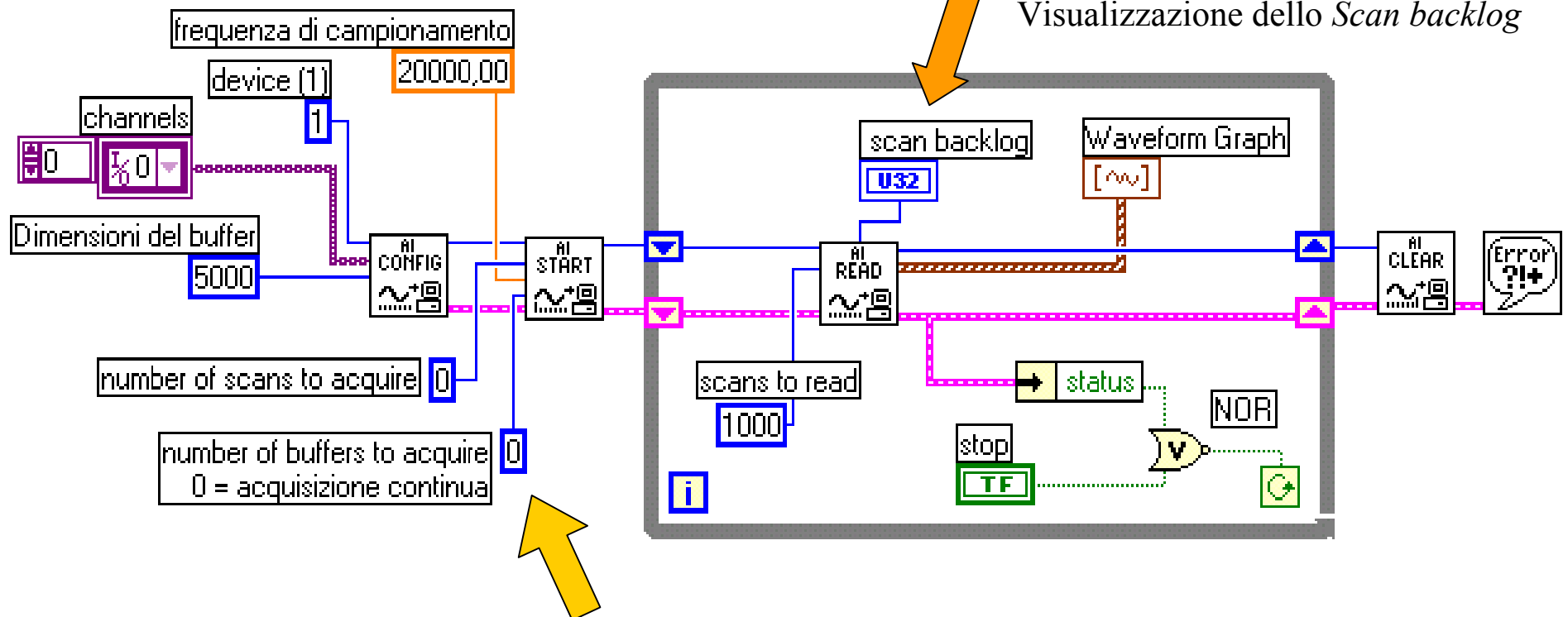
### Requisiti per l'acquisizione continua

- La velocità di lettura deve essere uguale o maggiore di quella di scrittura
- La velocità di scrittura dipende dalla frequenza di campionamento scelta
- La velocità di lettura dipende dal tempo necessario alla elaborazione di ogni sequenza di  $N_{\text{read}}$  punti

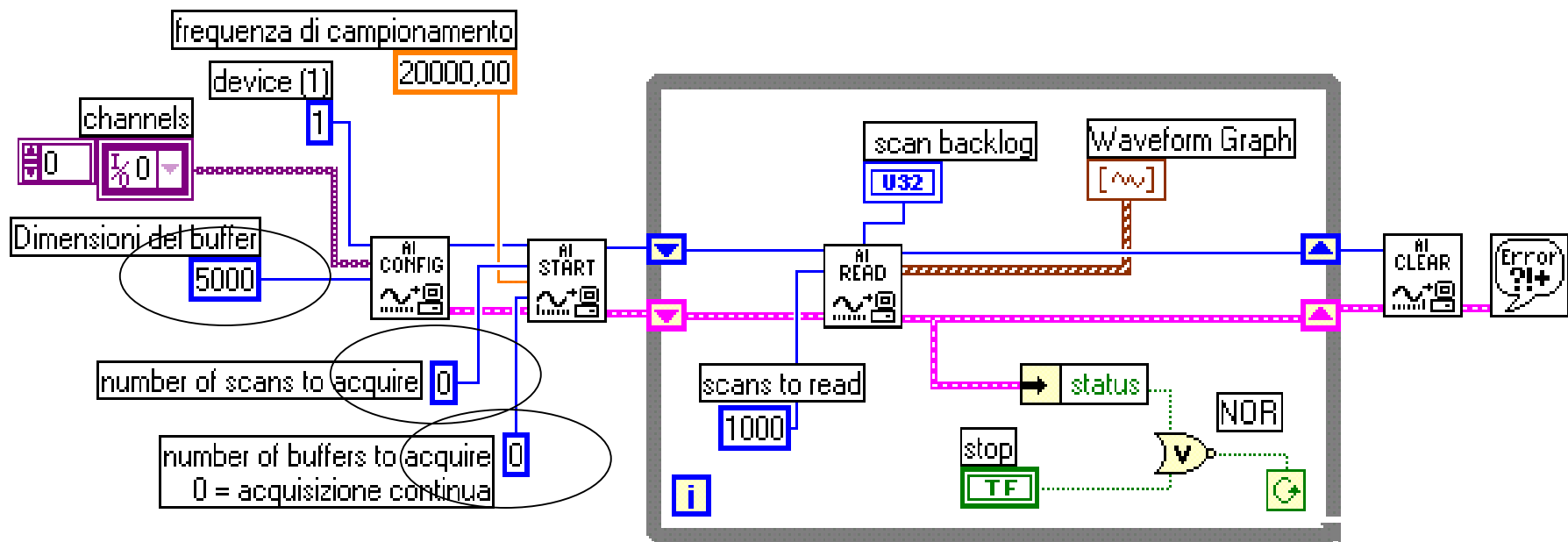


Se lo *scan backlog* supera  $N_{\text{buff}}$  vengono sovrascritti dati non ancora letti

# Esempio: acquisizione continua



# Acquisizione continua con funzioni di medio livello (VI Intermediate)

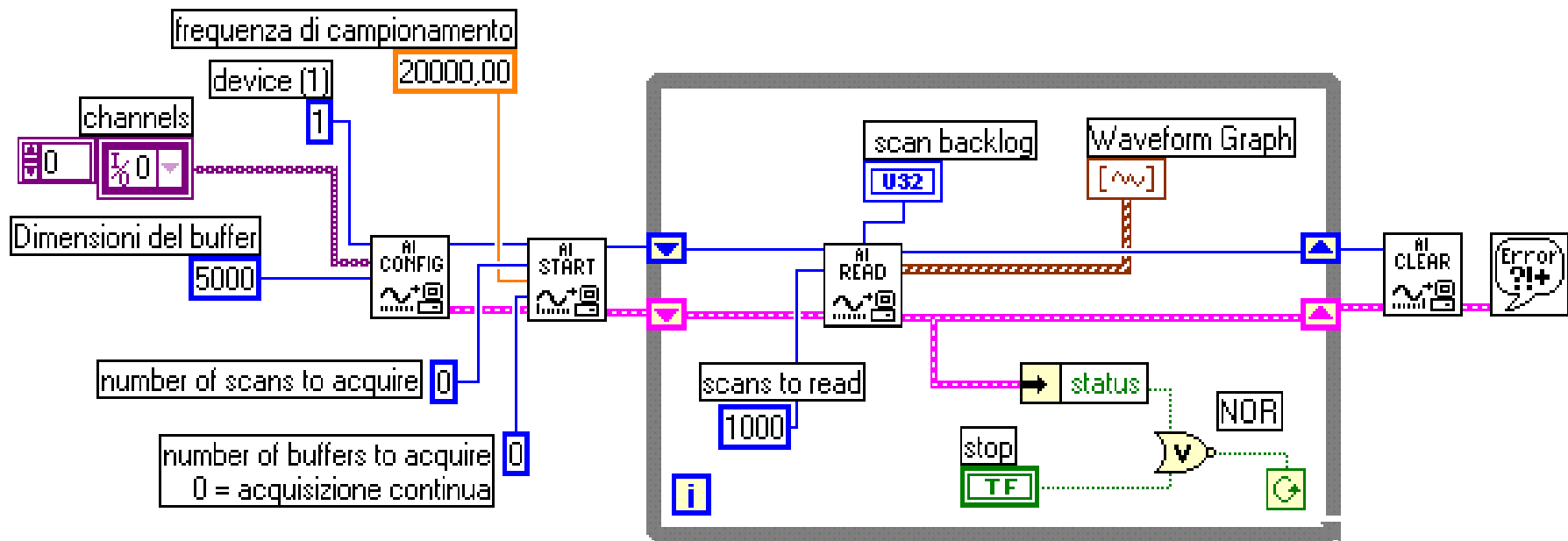


Il buffer del PC è allocato dal parametro **Dimensioni del buffer** del VI AI Config.

Impostando a zero **number of scans to acquire** nel VI AI Start si implementa un buffer circolare, mentre ponendo **number of buffers to acquire** a zero si richiede un'acquisizione continua.



# Acquisizione continua con funzioni di medio livello (VI Intermediate)

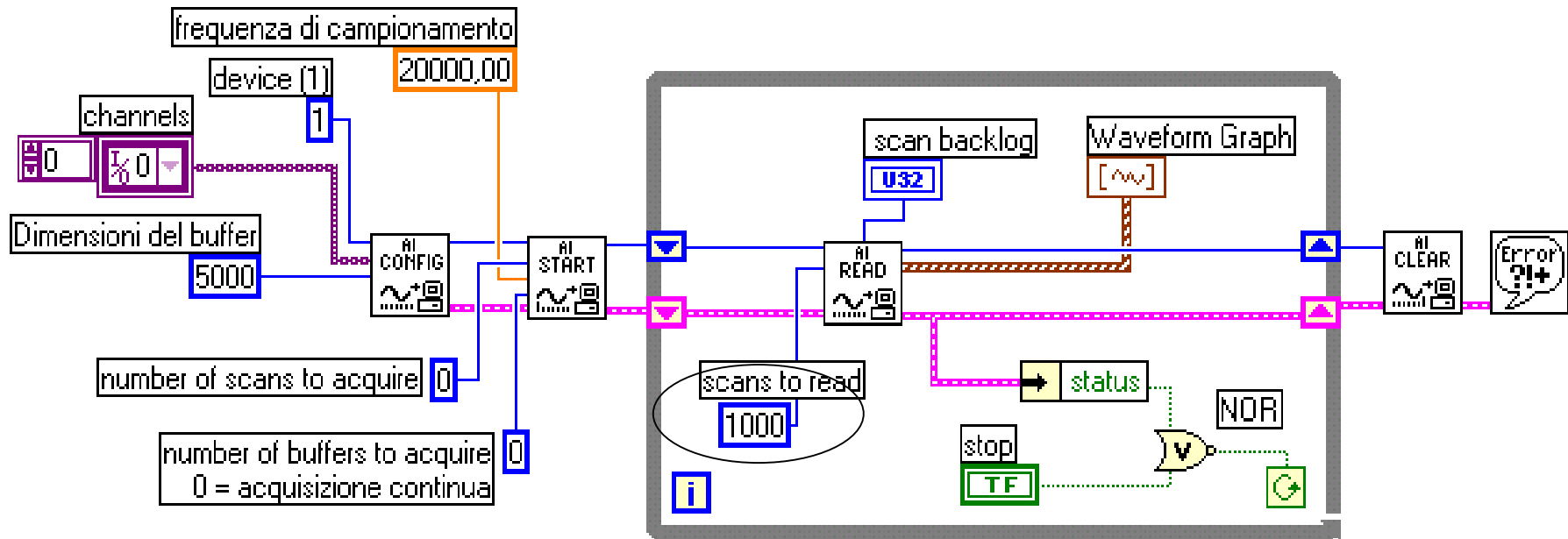


Il VI AI Read, inserito in un ciclo *While* si occupa di leggere in maniera ciclica i dati presenti nel buffer precedentemente allocato

La velocità con cui verrà letto il buffer dipenderà dal tipo e dal numero di operazioni eseguite all'interno del ciclo

Se tali operazioni non sono troppo onerose, il buffer verrà letto ancora abbastanza rapidamente e quindi verranno realmente elaborati tutti i dati senza perdita di nessuna informazione (a causa della sovrascrittura dei dati)

# Acquisizione continua con funzioni di medio livello (VI Intermediate)



È una buona pratica impostare **scans to read** ad un quarto o alla metà delle **dimensioni del buffer** per un'acquisizione bufferizzata continua

E' importante controllare il numero di scansioni non lette nel buffer per verificare se il buffer si sta svuotando abbastanza velocemente. Per tale motivo è utile monitorare la quantità **scan backlog**

# Acquisizione continua con funzioni di medio livello (VI Intermediate)

## ...ALCUNI CONSIGLI

L'errore più comune che si può incontrare quando si esegue un'acquisizione bufferizzata circolare è quello di sovrascrittura, codice di errore -10846.

Il problema si presenta quando LabVIEW non legge i dati dal buffer del PC abbastanza velocemente. Esistono diverse opzioni per evitare questo errore, ma non tutte le opzioni si possono adottare in tutte le situazioni e alcune funzionano meglio di altre.

1) Opzione: incrementare le dimensioni del buffer con il VI AI Config. Tuttavia, l'incremento delle dimensioni del buffer non risolve il problema se il buffer non viene svuotato abbastanza velocemente.

2) Opzione: Svuotare il buffer più rapidamente aumentando il numero di scansioni da leggere. Tuttavia, il numero di scansioni da leggere non può essere troppo elevato perché il VI AI Read attenderebbe fino a quando il numero di scansioni nel buffer non diventa uguale al numero di scansioni da leggere.

# Acquisizione continua con funzioni di medio livello (VI Intermediate)

## ...ALCUNI CONSIGLI

- 3) Opzione: diminuire la velocità di scansione (frequenza di campionamento) con il VI AI Start. Questa impostazione rallenta la velocità a cui i dati vengono inviati al buffer (vengono scritti), ma potrebbe non essere un'opzione valida se si vuole una certa velocità di scansione.
- 4) Opzione: si deve evitare di rallentare il ciclo con analisi ed elaborazioni non necessarie.