

Corso di laurea in Ingegneria delle Telecomunicazioni

Misure Elettroniche per le Telecomunicazioni II

LabView

Gianfranco Miele
g.miele@unicas.it

Le stringhe

- Una stringa è una sequenza di caratteri
- Ogni carattere ha una corrispondente codifica numerica in byte (8 bit) secondo lo standard ASCII
- I primi 32 dei 256 caratteri ASCII non sono visualizzabili (controllo)
- Gli ultimi 128 caratteri costituiscono il set *esteso*
- Impieghi: visualizzazione di messaggi, I/O su file, controllo di strumentazione
- Esistono controlli e indicatori di tipo stringa

<i>Stringa:</i>	V	D	C		-	1	2	.	4	5	6
<i># ASCII:</i>	86	68	67	32	45	49	50	46	52	53	54

Display mode degli indicatori

Normal display



\ code display



Backslash codes per alcuni caratteri di controllo:

`\b` backspace

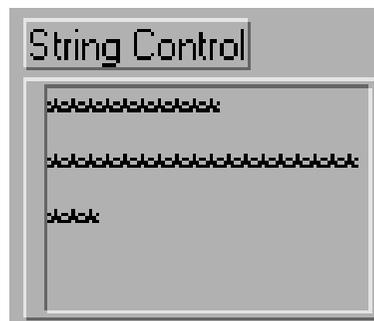
`\s` spazio

`\r` return (CR)

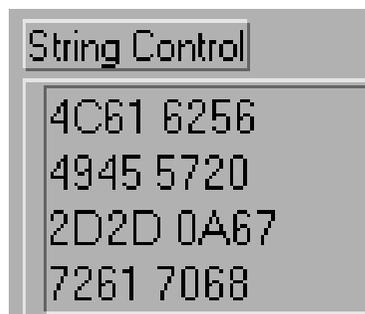
`\n` new line (LF)

`\t` tab

Password display



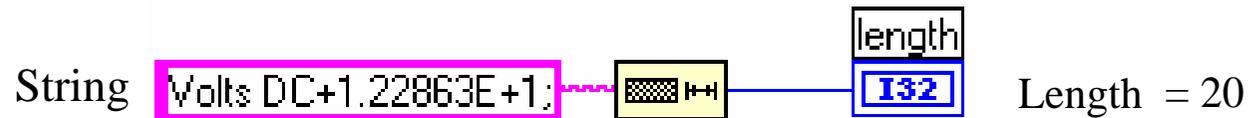
Hex display



Funzioni per le stringhe

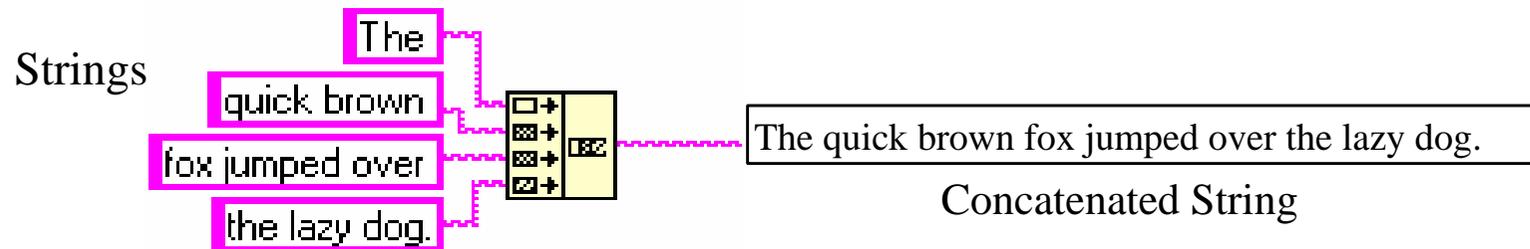
String Length

string  length



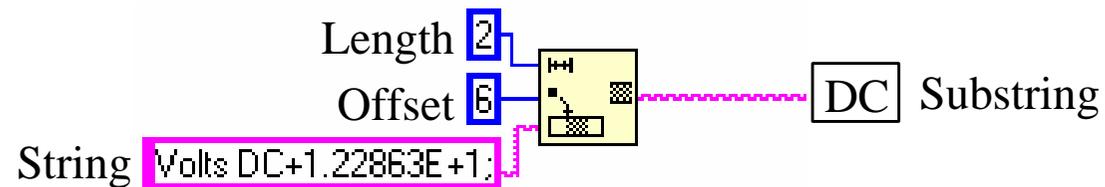
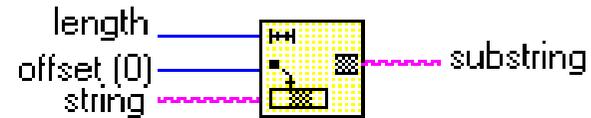
Concatenate Strings

string0  concatenate of string0, string1, ...
string1

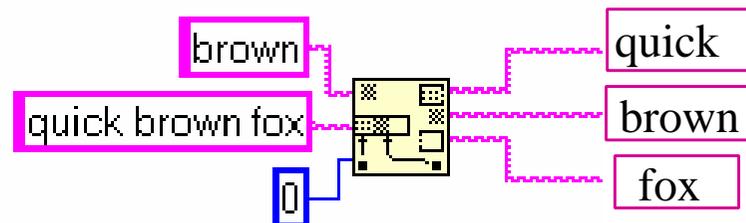
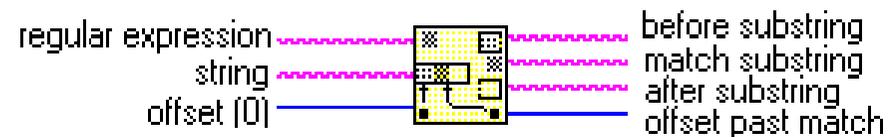


Funzioni per le stringhe

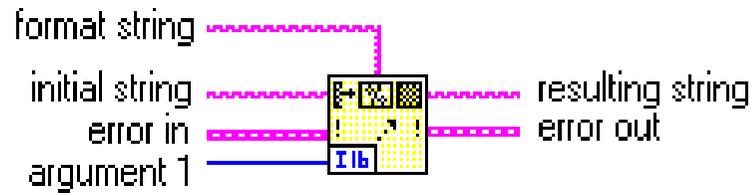
String Subset



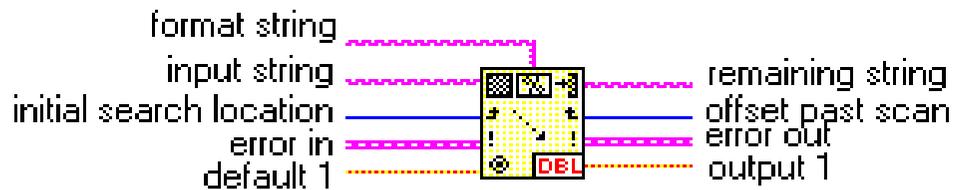
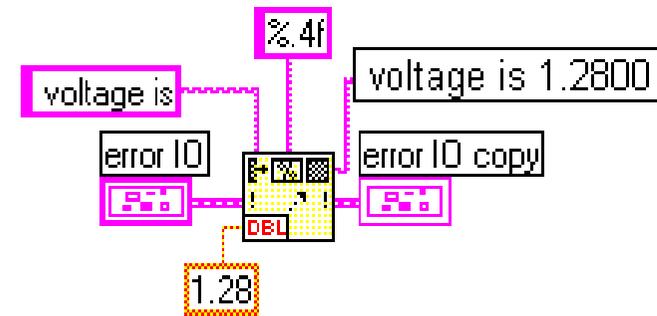
Match Pattern



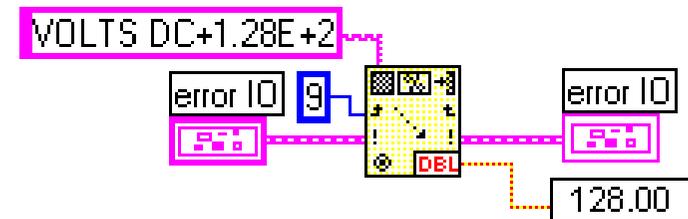
Funzioni per le stringhe



Format Into String
(è resizable)



Scan From String
(è resizable)

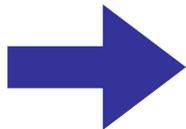


Stringhe di formato

- Indicano il formato, il tipo di dato, il numero di caratteri, l'allineamento ed il numero di cifre decimali (per i *floating point*) con cui visualizzare un dato
- Si usano come nella funzione **printf()** del linguaggio C

Esempio:

Risultato = %.1f

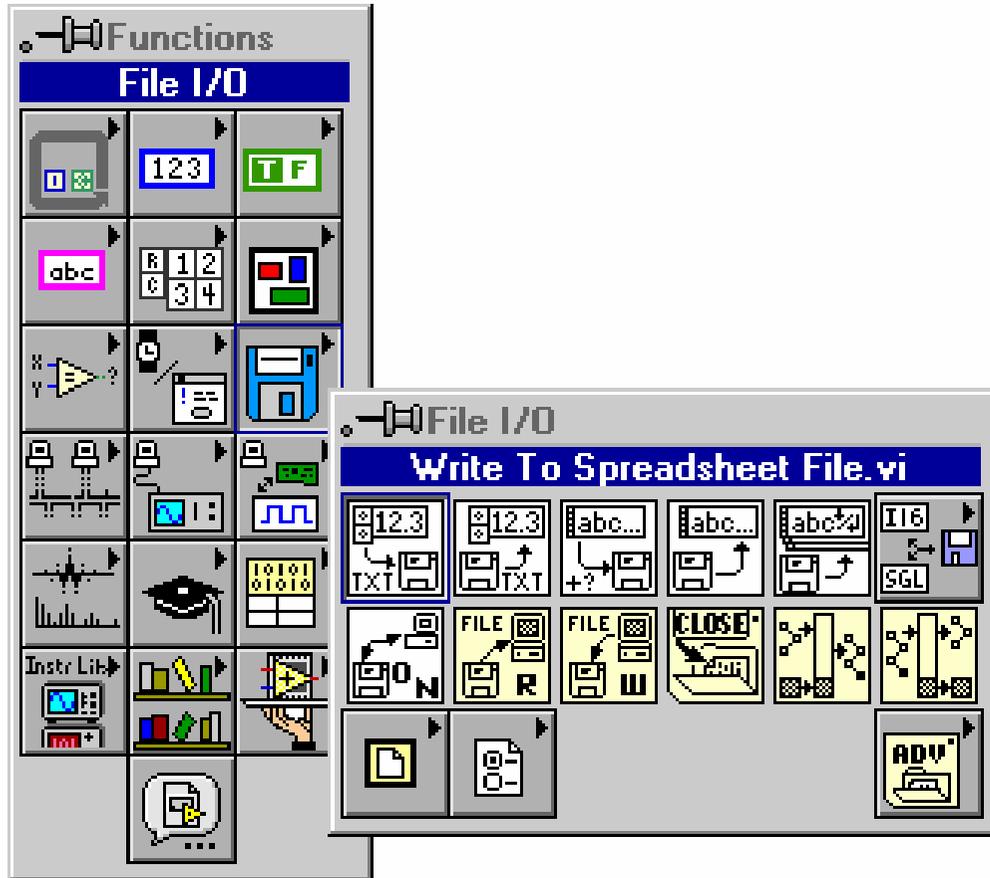


Risultato = 12.5

%d	intero con segno
%u	intero senza segno
%s	stringa
%f	floating point (15.012)
...	...

Dal menu di contesto, “*Edit Format String*” apre una finestra di dialogo che facilita la preparazione della stringa di formato

Funzioni per l'I/O su file



Alto livello:

- Read/Write to spreadsheet file
- Read/Write characters to file
- Read lines from file
- Read/Write to binary file

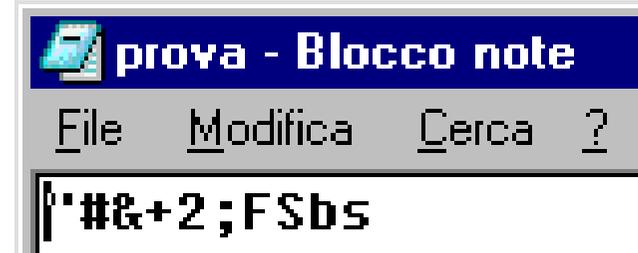
Livello intermedio:

Open, Read, Write, Close

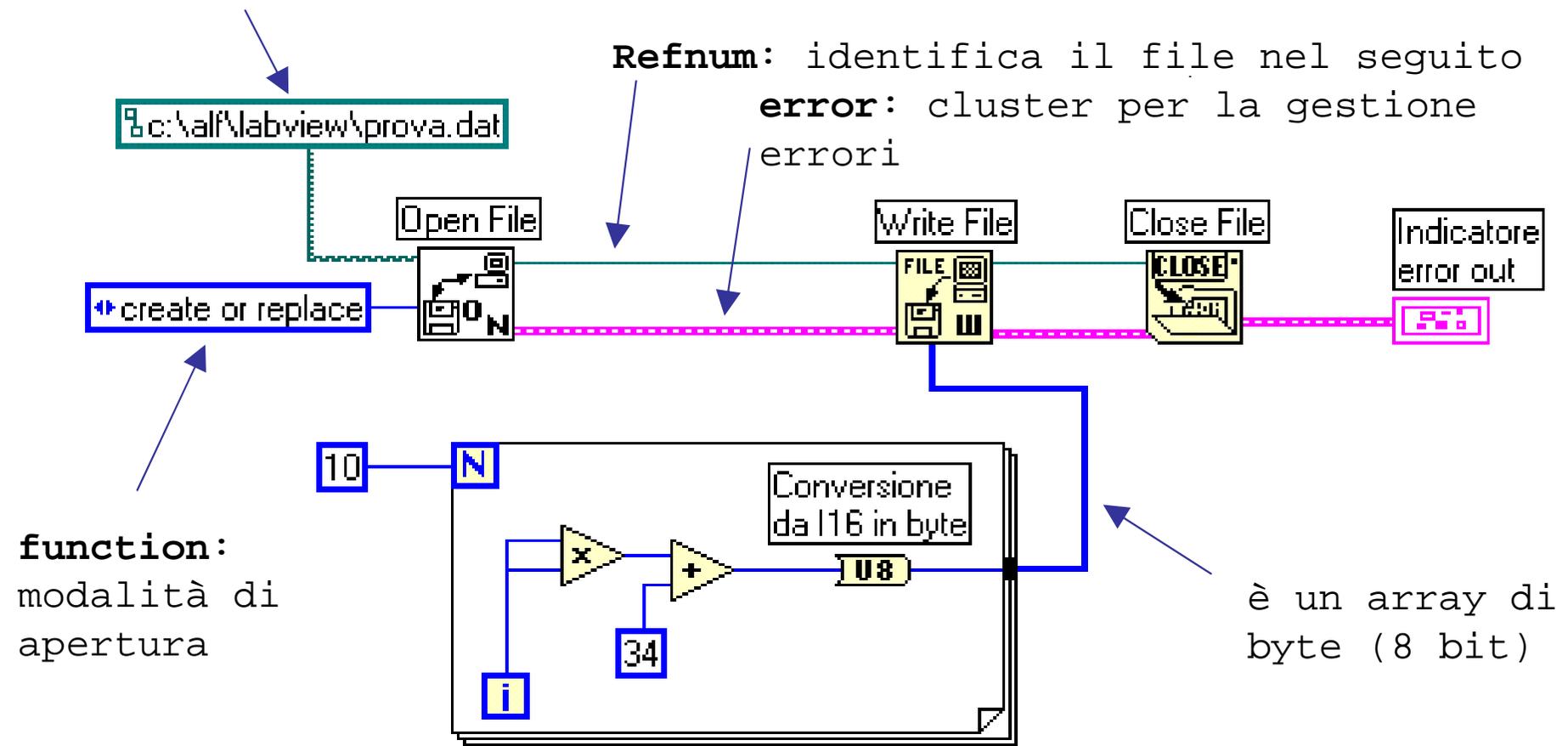
Livello avanzato:

- Gestione directory
- Dipendenti dal S.O.
- File dialog

Scrittura su file (funzioni di livello intermedio)

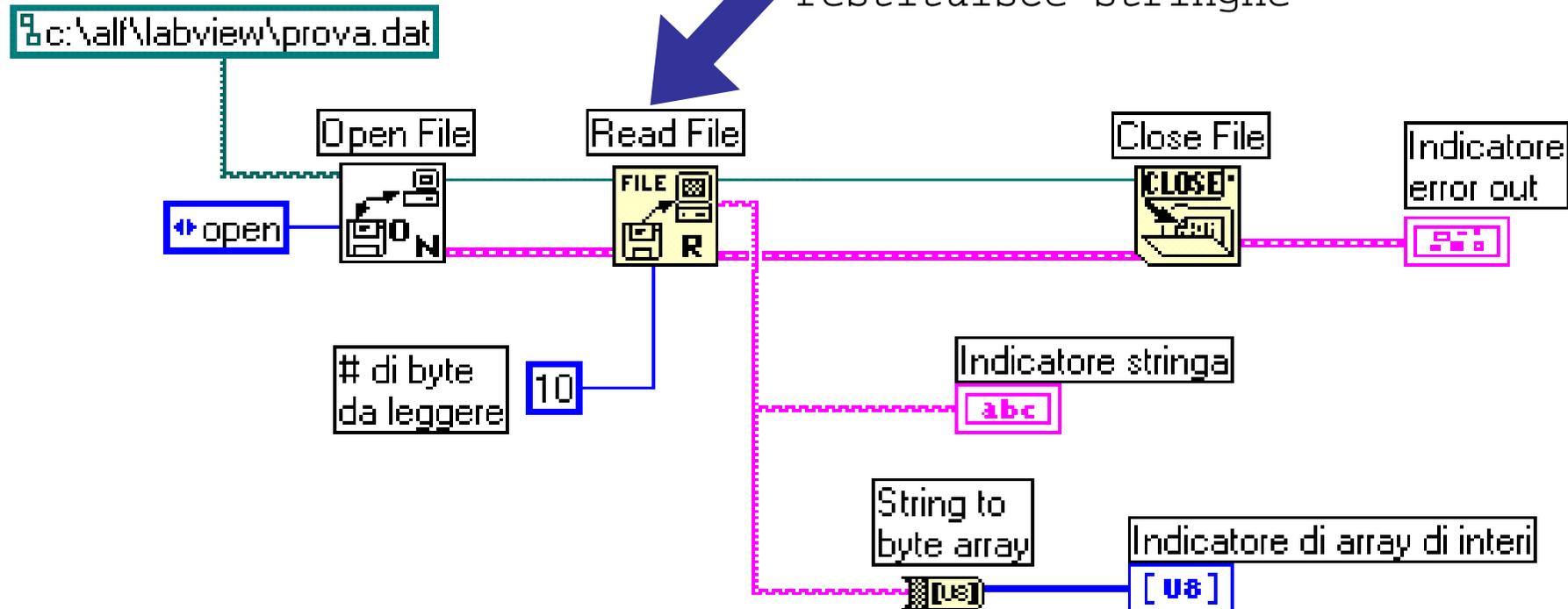


File path: il formato dipende dal s.o.



Letture da file (funzioni di livello intermedio)

La *Read File* per default restituisce stringhe



Indicatore di array di interi

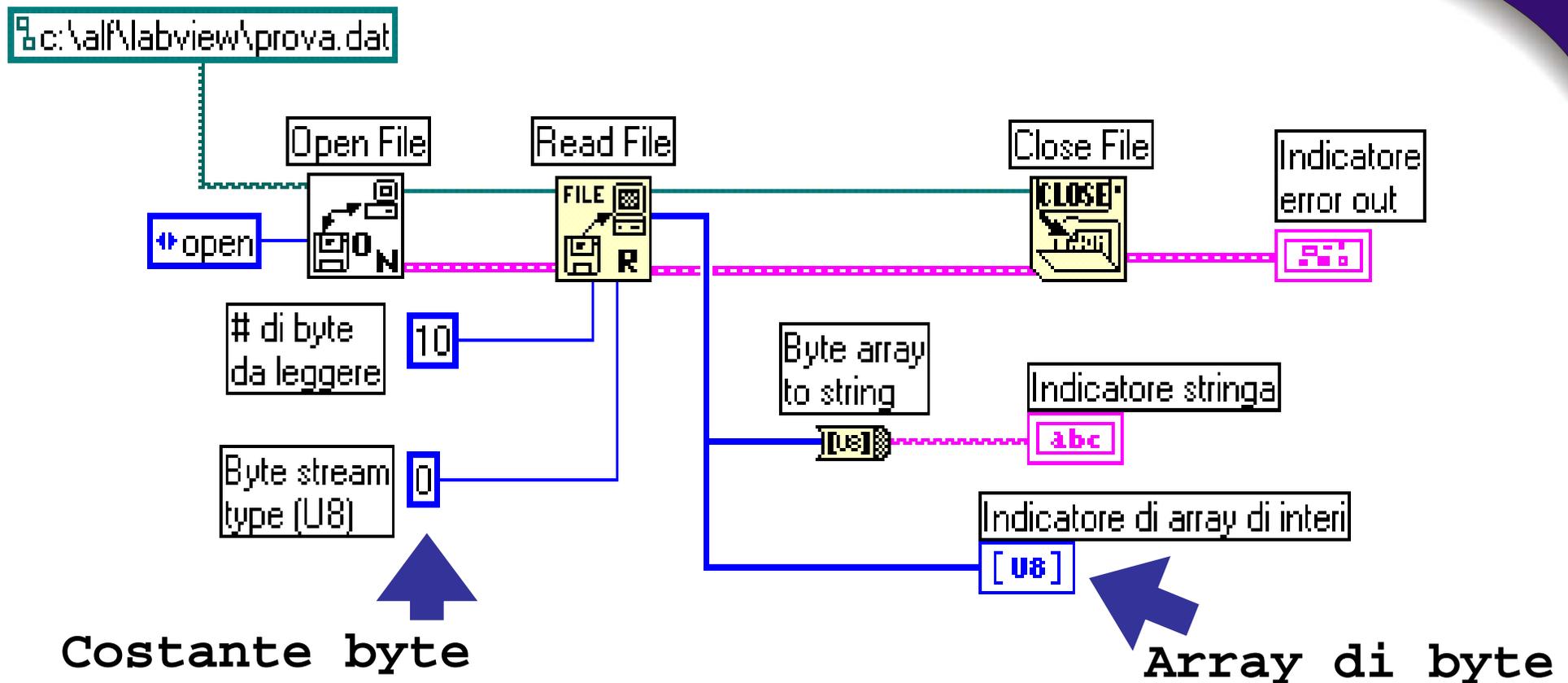
0	34	35	38	43	50	59	70	83	98	115
---	----	----	----	----	----	----	----	----	----	-----

Indicatore stringa

```
"#&+2;FSbs"
```

Il contenuto di un file può avere diverse rappresentazioni

Letture da file (funzioni di livello intermedio)

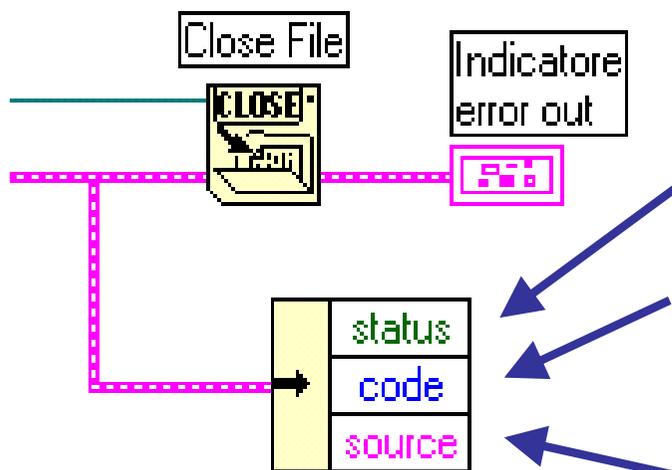


Il tipo di dato connesso al *Byte stream type* fissa il tipo degli elementi dell'array in uscita dalla *Read File*

Analogamente, la *Write File* è una funzione polimorfica

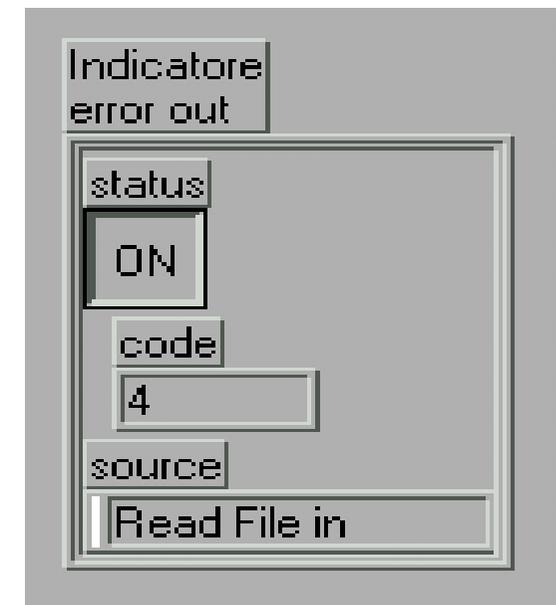
Gestione degli errori

- Non appena chiamata, ogni funzione controlla il connettore **error in**;
- Se trova **status = True**, significa che a monte si è verificato un errore: non esegue alcuna operazione e termina;
- Se si verifica un errore durante la sua esecuzione, **error out = True**.



Error Cluster:

- C'è stato errore?
- Numero associato all'errore
- In quale VI si è verificato?



File di testo e file binari

Un file di testo è una sequenza di stringhe di caratteri ASCII delimitate da sequenze di *fine linea* (CR+LF):

a = 1.234e-5<CR><LF>b = 200<CR><LF>z=FALSE ...

- Facilmente interpretabile
- Scambio dati con altre applicazioni (fogli elettronici o *word processor*)
- La rappresentazione dei valori numerici è a lunghezza variabile
- L'accesso ai dati deve essere sequenziale

In un file binario la codifica dei dati dipende dalla particolare applicazione

- La rappresentazione dei valori numerici è a lunghezza fissa, e richiede un numero minore di byte
- L'accesso può essere casuale

```
a = 1.234e-5  
b = 200  
z=FALSE
```

Spreadsheet file

- I dati sono organizzati in righe e colonne
- Le colonne sono separate da un delimitatore (TAB)
- Le righe sono separate da sequenze *new line* (CR+LF)

File:

```
0<TAB>0.4258<CR><LF>  
1<TAB>0.3073<CR><LF>  
2<TAB>0.9453<CR><LF>  
3<TAB>0.964<CR><LF>  
4<TAB>0.9517<CR><LF>
```

Nel foglio elettronico:

	A	B	C
1	0	0.4258	
2	1	0.3073	
3	2	0.9453	
4	3	0.964	
5	4	0.9517	
6			

Spreadsheet file



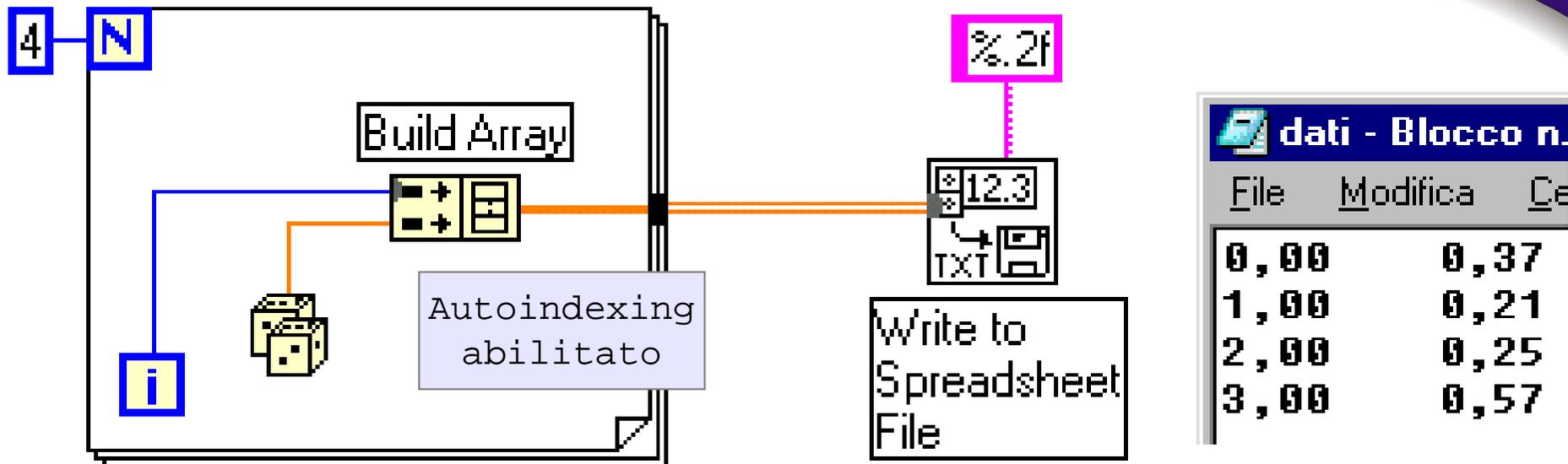
- **Write to Spreadsheet File**
- **Read from Spreadsheet File**
- **Write Characters to File**
- **Read Characters from File**
- **Read Lines from File**

Effettuano direttamente le operazioni più frequenti di I/O su file di testo

Aprono e chiudono il file (non deve farlo il chiamante)

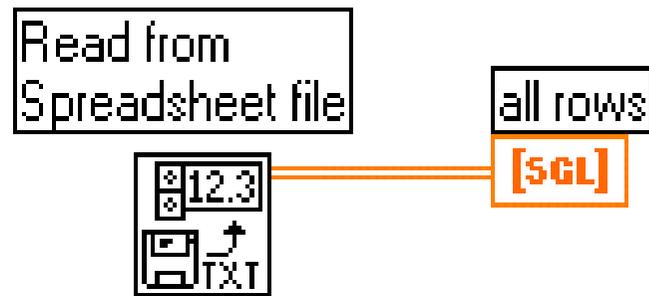
Nel loro diagramma a blocchi chiamano le funzioni di livello intermedio

Scrittura su spreadsheet file



- La funzione *Write to Spreadsheet File* accetta in ingresso array 1D e 2D
- Se non specifica un nome di file, si apre una finestra di dialogo “Apri file..”
- Si può specificare il carattere delimitatore (TAB per default), trasporre l'array 2D, aggiungere i dati ad un file esistente (*append*)

Lettura su spreadsheet file



0	0,000	0,350
0	1,000	0,190
	2,000	0,280
	3,000	0,290

Altre funzioni di alto livello:

- *Write Characters to File*: Scrive una stringa di caratteri in un file
- *Read Lines from Files*: Legge un numero specificato di *linee* da un file
- *Binary File VIs*: 4 funzioni per la lettura/scrittura di array di **I16** o **SGL** su file