

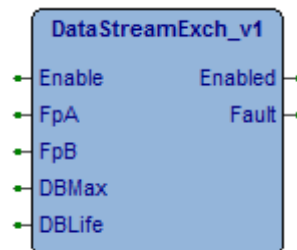
7.17.8 DataStreamExch_v1, exchanges data between two I/O streams

Type	Library
Function	eLLabNetworkLib_A600

Questo blocco funzione esegue lo scambio data tra due streams di I/O. Definendo uno stream come socket TCP e l'altro come seriale è possibile realizzare un convertitore Ethernet/Seriale.

In **FpA** ed **FpB** occorre indicare i file pointers degli streams di I/O. Il FB riceve i dati da uno stream e li invia sull'altro stream. Per il trasferimento dei dati viene allocato dinamicamente un buffer di memoria che può raggiungere al massimo la dimensione definita in **DBMax** (Si consiglia 2048 bytes).

Se non c'è comunicazione tra gli streams dopo il tempo definito in **DBLife** (Si consiglia 5 secondi) il buffer di memoria viene disallocato liberando risorse di sistema. Se valore impostato è 0 il buffer non viene mai disallocato.



Enable (BOOL)	Abilitazione blocco funzione.
FpA (FILEP)	Flusso dati stream di comunicazione.
FpB (FILEP)	Flusso dati stream di comunicazione.
DBMax (UINT)	Dimensione massima allocazione buffer dati (Bytes). Valore massimo 4096.
DBLife (REAL)	Tempo di vita buffer dati (S).
Enabled (BOOL)	Blocco funzione abilitato.
Fault (BOOL)	Attivo per un loop di programma in caso di errore.

Codici di errore

In caso di errore si attiva l'uscita **Fault**, con [SysGetLastError](#) è possibile rilevare il codice di errore.

10048100 Errore valore DBMax.

Esempi

Semplice convertitore Ethernet/Seriale, accetta connessione TCP su porta 2000 e utilizza la porta seriale COM0.

Definizione variabili

	Name	Type	Address	Array	Init value	Attribute	Description
1	Fp	FILEP	Auto	No		..	File pointer
2	Sp	SysSerialPort	Auto	No		..	Serial port management
3	Sk	SysTCPServer	Auto	No		..	Socket server
4	DExch	DataStreamExch_v1	Auto	No		..	Data exchange

Esempio ST (PTP119B400, ST_DataStreamExch_v1)

```

(* ----- *)
(* INITIALIZATIONS *)
(* ----- *)
(* Program initializations. *)

IF (SysFirstLoop) THEN

    (* Initialize serial port. *)

    Sp.COM:=ADR('COM0'); (* Serial port *)
    Sp.Baudrate:=115200; (* Baud rate *)
    Sp.Parity:='E'; (* Parity *)
    Sp.DataBits:=8; (* Data bits *)
    Sp.StopBits:=1; (* Stop bits *)
    Sp.DTRManagement:=DTR_AUTO_WO_TIMES; (* DTR management *)

    (* Initialize socket server. *)

    Sk.FilesArr:=ADR(Fp); (* File array *)
    Sk.MaxConn:=1; (* Number of connections accepted *)
    Sk.LocalAdd:=ADR('0.0.0.0'); (* Local address *)
    Sk.LocalPort:=2000; (* Local port *)
    Sk.LifeTm:=60; (* TSocket life time (S) *)
    Sk.FlushTm:=10; (* Socket flush time (mS) *)
    Sk.RxSize:=1024; (* Rx size buffer *)
    Sk.TxSize:=1024; (* Tx size buffer *)

    (* Initialize data stream exchange . *)

    DExch.DBMax:=2048; (* Data buffer maximum size *)
    DExch.DBLife:=5.0; (* Data buffer life (S) *)
END_IF;

(* ----- *)
(* DATA STREAM EXCHANGE *)
(* ----- *)
(* Execute the data stream exchange. *)

Sp(Open:=TRUE); (* Serial port management *)
Sk(Enable:=TRUE); (* TCP server management *)
DExch.FpA:=Sp.File; (* File pointer *)
DExch.FpB:=Fp; (* File pointer *)
DExch(Enable:=(Sk.ConnPeers <> 0)); (* Data exchange *)

(* [End of file] *)

```