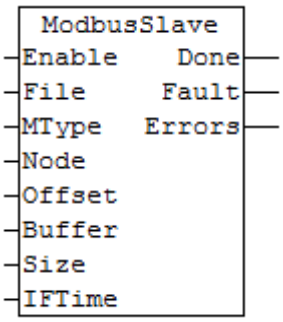
1.1.2 ModbusSlave, modbus slave

Type	Library
FB	eCDSUtilityLib_A100



Questo blocco funzione esegue la gestione del protocollo modbus slave con ***MType*** è possibile selezionare il tipo di protocollo RTU, Ascii ed over IP. Con ***File*** è possibile definire il terminale di I/O su cui effettuare la comunicazione.

Occorre definire il nodo modbus ***Node***, e l'eventuale offset di indirizzo frame modbus ***Offset***. I comandi modbus ricevuti operano sul buffer di memoria il cui indirizzo è definito in ***Buffer*** e la dimensione in bytes è definita in ***Size***.

In ***IFTime*** occorre definire il tempo di interframe dei comandi modbus, cioè il tempo che intercorre tra la ricezione di un comando ed il comando successivo. Su linea seriale questo tempo coincide con il tempo di ricezione di 3 caratteri al baud rate definito.

Alla ricezione di ogni comando modbus corretto si attiva per un loop l'uscita ***Done***, in caso di errore comando viene attivata per un loop l'uscita ***Fault*** ed incrementato il valore in ***Errors***.

Enable (BOOL) Comando di abilitazione blocco funzione.

File (FILEP) Flusso dati ***stream*** ritornato dalla funzione ***Sysfopen***.

MType (USINT) Tipo di protocollo modbus. 0:RTU, 1:Ascii, 2:TCP

Node (USINT) Numero di nodo modbus (Range da 0 a 255).

Offset (UINT) Offset su indirizzo modbus ricevuto nel frame dati (Range da 16#0000 a 16#FFFF).

Buffer (@USINT) Indirizzo buffer dati su cui operano i comandi modbus.

Size (UINT) Dimensione in byte del buffer dati su cui operano i comandi modbus.

IFTime (UDINT) Tempo ricezione caratteri (µS), se comunicazione su porta seriale il tempo deve essere definito in base al baud rate. Nel caso di comunicazione su rete ethernet è possibile definire il valore minimo.

Baud rate	Tempo
300	112000
600	56000
1200	28000
2400	14000
4800	7000
9600	3430

Baud rate	Tempo
19200	1720
38400	860
57600	573
76800	429
115200	286

Done (BOOL) Attivo per un loop alla ricezione di comando modbus.

Fault (BOOL) Attivo per un loop su errore ricezione comando modbus.

Errors (UDINT) Numero di errori riscontrati. Viene incrementato ad ogni nuovo errore, raggiunto il valore massimo il conteggio riparte da 0.

Comandi supportati

Il blocco funzione supporta solo alcuni comandi previsti dal protocollo modbus, i comandi supportati sono:

Codice	Descrizione
16#01	Read coil status (Massimo 250 coils)
16#02	Read input status (Massimo 250 coils)
16#03	Read holding registers (Massimo 125 registri)
16#04	Read input registers (Massimo 125 registri)
16#05	Force single coil
16#06	Preset single register
16#08	Loopback diagnostic test
16#0F	Force multiple coils (Massimo 250 coils)
16#10	Preset multiple registers (Massimo 125 registri)
16#41	Read memory bytes (User function) (Massimo 250 bytes)
16#42	Write memory bytes (User function) (Massimo 250 bytes)

Codici di errore

In caso di errore si attiva l'uscita ***Fault***, con ***SysGetLastError*** è possibile rilevare il codice di errore. In caso di eccezione su comando modbus viene riportato il codice di errore ma non viene attivata l'uscita ***Fault***.

10038010 Valore di ***File*** non definito.

10038050 Errore di timeout esecuzione.

10038060 Errore esecuzione.

10038080 Errore definizione ***Type***.

10038100~10 Errore in ricezione frame (Carattere errato, lunghezza errata).

10038200~2 Errore trasmissione frame risposta.

10038501 Eccezione 01.***Illegal function***, comando ricevuto non è tra quelli gestiti.

10038502 Eccezione 02. ***Illegal data address***, comando ricevuto ha indirizzo o numero dati fuori range.

10038503 Eccezione 03. ***Illegal data value***, comando ricevuto ha campo dati fuori range.

10038504 Eccezione 04.***Failure in associated device***, comando ricevuto contiene imprecisioni.

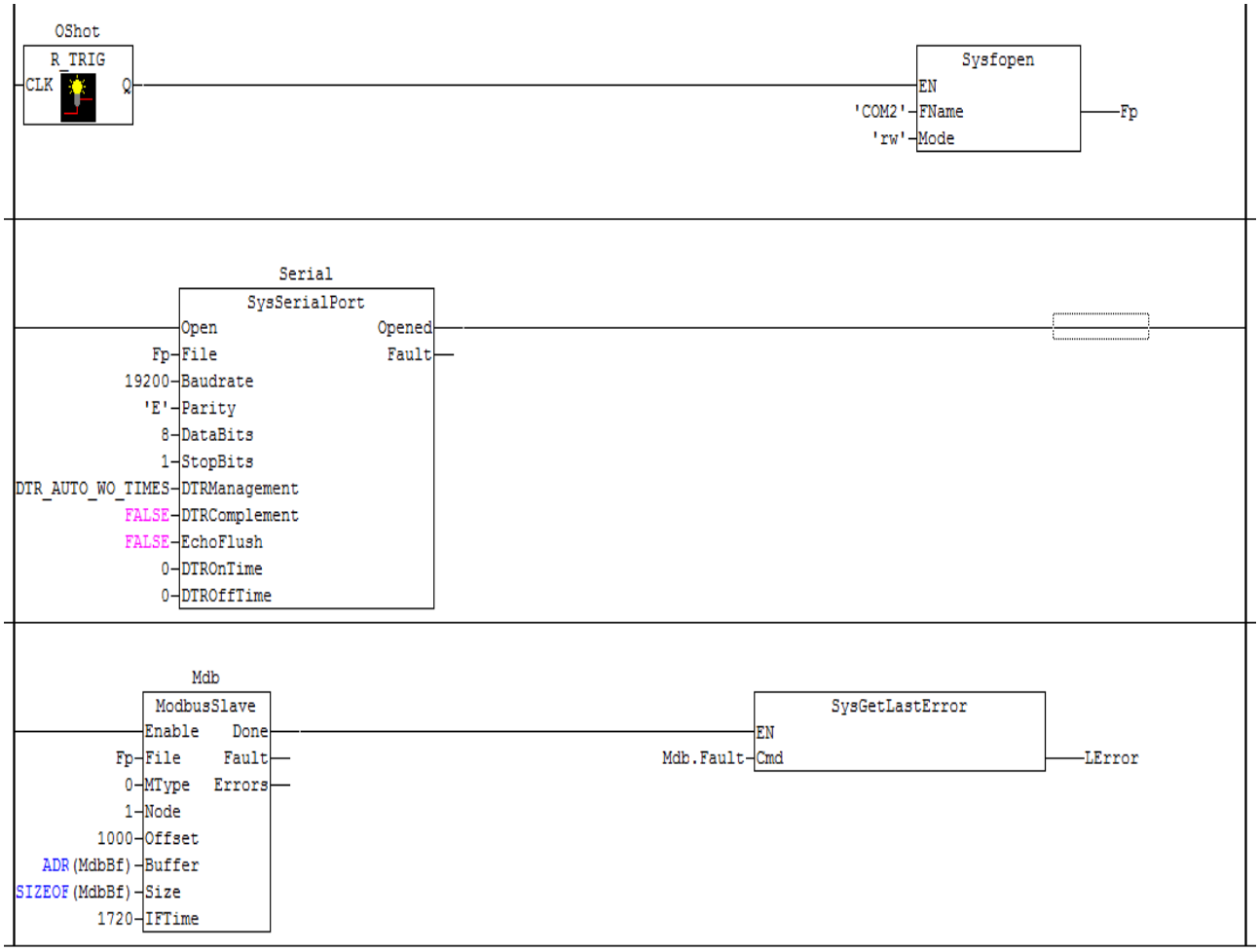
Esempi

Viene gestito il protocollo modbus slave su porta seriale ***COM2***, i comandi modbus possono agire sull'array ***MdbBf***.

Definizione variabili

```
VAR
  Serial: SysSerialPort; (* Serial port *)
  TCPSkt: SysTCPServer; (* TCP/IP socket *)
  Mdb: ModbusSlave; (* Modbus slave *)
  Fp: FILEP; (* File pointer *)
  Enable: BOOL; (* Abilitazione *)
  OShot: R_TRIG; (* One shot *)
  MdbBf: ARRAY [0..31] OF UINT; (* Buffer modbus *)
  LError: UDINT;
END_VAR
```

Esempio LD *(PTP114A610, LD_ModbusSlave)*



Esempi

Viene gestito il protocollo modbus slave su due socket TCP, i comandi modbus possono agire sull'array ***MdbBf***.

Definizione variabili

```
VAR
  Init: BOOL:=TRUE; (* Init command *)
  i: USINT; (* Auxiliary counter *)
  Fp: ARRAY [0..1] OF FILEP; (* File pointer *)
  PSts: ARRAY [0..1] OF SysGetIpInfos; (* Peer status *)
  Svr: SysTCPServer; (* TCP server *)
  Mdb: ARRAY [0..1] OF ModbusSlave; (* Modbus slave *)
  MdbBf: ARRAY [0..31] OF UINT; (* Buffer modbus *)
END_VAR
```

Esempio ST *(PTP114A610, LD_ModbusSlave)*

```
  (* ---------------------------------------------------------------------- *)
  (* INIZIALIZZAZIONE                                                       *)
  (* ---------------------------------------------------------------------- *)
  (* Open the files and configure the TCP server. *)

  IF (Init) THEN
    Init:=FALSE; (* Init command *)
    FOR i:=0 TO 1 DO Fp[i]:=Sysfopen('TCPSKT', 'rw'); END_FOR; (* File pointer *)

    (* Eseguo configurazione TCP server. *)

    Svr.Enable:=TRUE; (* Enable command *)
    Svr.Files:=ADR(Fp); (* File pointer *)
    Svr.PeerIPEn:='192.168.0.255'; (* Accept all IPs in the range 192.168.0.xxx *)
    Svr.PeerPortEn:=16#FFFF; (* Accept all ports *)
    Svr.Port:=502; (* Listening port *)
    Svr.MaxConn:=2; (* Number of accepted connections *)
    Svr.LifeTm:=30; (* Life time (S) *)

    (* Eseguo configurazione Modbus slave. *)

    FOR i:=0 TO 1 DO
      Mdb[i].File:=Fp[i]; (* Terminal I/O pointer *)
      Mdb[i].MType:=2; (* Modbus type *)
      Mdb[i].Node:=1; (* Node number *)
      Mdb[i].Offset:=1000; (* Modbus address offset *)
      Mdb[i].Buffer:=ADR(MdbBf); (* Data buffer *)
      Mdb[i].Size:=SIZEOF(MdbBf); (* Data buffer size *)
      Mdb[i].IFTime:=0; (* Interframe time (uS) *)
    END_FOR;
  END_IF;

    (* ---------------------------------------------------------------------- *)
    (* GESTIONE MODBUS SLAVE                                                  *)
    (* ---------------------------------------------------------------------- *)
  (* Gestione server sockets. *)

  Svr(); (* Manage the TCP/IP server *)
  FOR i:=0 TO 1 DO PSts[i](File:=Fp[i]); END_FOR; (* Peer status *)
  FOR i:=0 TO 1 DO Mdb[i](Enable:=SysIsFOpen(Fp[i])); END_FOR;
```