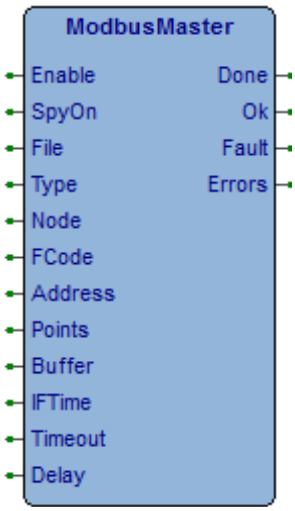
1.1.3 ModbusMaster, modbus master

Type	Library
FB	eLLabUtyLib_C030



Questo blocco funzione esegue la gestione del protocollo modbus master, con ***Type*** 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.

Attivando ***Enable*** sul terminale di I/O viene inviato un frame per eseguire la funzione modbus definita in ***Fcode*** sul nodo definito in Node. Terminata l'esecuzione del comando viene attivata l'uscita ***Done***. Se l'esecuzione comando ha esito positivo si attiva per un loop di programma l'uscita ***Ok***. Disattivando ***Enable*** si azzera ***Done*** e l'eventuale ***Fault***, per eseguire nuovamente il comando occorre riabilitare l'ingresso ***Enable***. L'ingresso ***SpyOn*** se attivo permette di spiare il funzionamento della FB.

Se ***FCode*** è una funzione di lettura, il valore delle variabili a partire dall'indirizzo definito in ***Address*** per il numero di variabili definito da ***Points***, viene letto dal sistema slave e trasferito nelle variabili indirizzate da ***Buffer***.

Se ***FCode*** è una funzione di scrittura, il valore delle variabili presenti nel buffer di memoria indirizzato da ***Buffer*** per il numero di variabili definito da ***Points***, è inviato al dispositivo slave che lo trasferirà nelle sue variabili a partire dall'indirizzo definito in ***Address***.

In caso di errore esecuzione o tempo di esecuzione comando superiore al tempo definito in ***Timeout***, viene attivata per un loop di programma l'uscita ***Fault*** ed incrementato il valore in ***Errors***.

Enable (BOOL) Comando abilitazione esecuzione comando modbus. Per rieseguire il comando disabilitare e poi riabilitare questo ingresso.

SpyOn (BOOL) Se attivo permette di spiare il funzionamento della FB.

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

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

Node (USINT) Numero di nodo modbus su cui effettuare il comando (Range da 0 a 255).

FCode (USINT) Codice funzione modbus da eseguire nel comando (Range da 0 a 255).

Codice	Descrizione
16#01	Read coil status (Massimo 250 coils)
16#02	Read input status (Massimo 125 inputs)
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#0F	Force multiple coils (Massimo 250 coils)
16#10	Preset multiple registers (Massimo 32 registri)
16#41	Read memory bytes (User function) (Massimo 250 bytes)
16#42	Write memory bytes (User function) (Massimo 250 bytes)

Address (UINT) Indirizzo di allocazione variabili su sistema slave. In accordo alle specifiche modbus l'indirizzo inviato nel frame dati è (***Address-1***) (Range da 16#0001 a 16#FFFF).

Points (USINT) Numero di variabili consecutive su cui opera il comando.

Buffer (@USINT) Indirizzo buffer dati letti o da scrivere.

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

Timeout (UINT) Tempo massimo esecuzione comando espresso in mS. Se il comando non termina nel tempo definito viene abortito ed attivata l'uscita ***Fault***.

Delay (UINT) Tempo di pausa dopo l'esecuzione del comando modbus espresso in mS.

Done (BOOL) Si attiva al termine della esecuzione comando.

Ok (BOOL) Attivo per un loop se esecuzione comando corretta.

Fault (BOOL) Attivo per un loop se errore esecuzione comando.

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

Trigger di spy

Se ***SpyOn*** attivo viene eseguita la funzione ***SysSpyData*** che permette di spiare il funzionamento della FB. Sono previsti vari livelli di triggers.

TFlags	Descrizione
16#00000001	***Tx:*** Invio frame comando modbus.
16#00000002	***Rx:*** Ricezione frame risposta modbus.

Codici di errore

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

10007010	Valore di ***File*** non definito.
10007050	Timeout esecuzione.
10007060	Errore esecuzione.
10007080	Errore definizione ***Type***.
10007100	Codice funzione definito in ***Function*** non gestito.
10007120	Valore di ***Points*** errato.
10007200~2	Errore trasmissione frame comando.
10007500~7	Errore in ricezione frame risposta (Carattere errato, lunghezza errata, CRC).

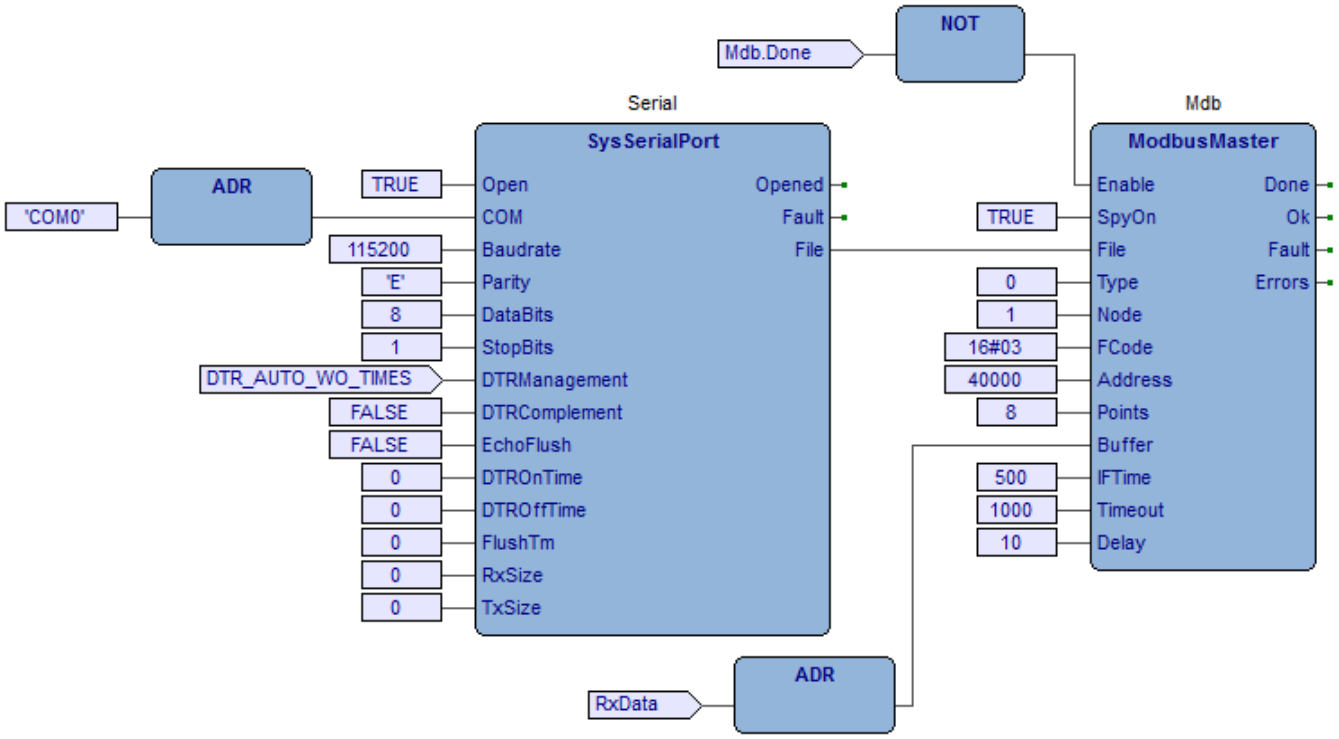
Esempi Modbus master su seriale

Viene presentato un esempio di lettura da un sistema SlimLine slave. Viene eseguita la lettura di 8 registri a partire da indirizzo 40000 dal nodo modbus 1. Il valore dei registri letti è trasferito nell'array ***RxData***. Terminata la lettura si attiverà per un loop l'uscita logica ***Done***.

Definizione variabili

	Name	Type	Address	Array	Init value	Attribute	Description
1	RxData	UINT	Auto	[0..7]		..	Rx data buffer
2	Mdb	ModbusMaster	Auto	No		..	Modbus master FB
3	Serial	SysSerialPort	Auto	No		..	Serial port

Esempio FBD *(PTP114A670, FBD_ModbusMasterSerial)*



Esempio ST *(PTP114A660, ST_ModbusMaster)*

```
(* Program initialization. *)

IF (SysFirstLoop) THEN

  (* Serial port initialization. *)

  Serial.COM:=ADR('COM0'); (* COM port definition *)
  Serial.Baudrate:=9600; (* Baudrate *)
  Serial.Parity:='N'; (* Parity *)
  Serial.DataBits:=8; (* Data bits *)
  Serial.StopBits:=1; (* Stop bits *)
  Serial.DTRManagement:=DTR_AUTO_WO_TIMES; (* DTR management *)
  Serial.DTRComplement:=FALSE; (* DTR complement *)
  Serial.EchoFlush:=FALSE; (* Received echo flush *)
  Serial.DTROnTime:=0; (* DTR On time delay (mS) *)
  Serial.DTROffTime:=0; (* DTR Off time delay (mS) *)
  Serial.FlushTm:=0; (* Flush time (mS) *)
  Serial.RxSize:=0; (* Rx buffer size *)
  Serial.TxSize:=0; (* Tx buffer size *)

  (* Modbus master initialization. *)

  Mdb.SpyOn:=TRUE; (* Spy On *)
  Mdb.Type:=0; (* Modbus type *)
  Mdb.Node:=1; (* Device modbus node *)
```

```
    Mdb.FCode:=16#03; (* Modbus function code *)
    Mdb.Address:=4000; (* Modbus register address *)
    Mdb.Points:=8; (* Modbus register points *)
    Mdb.Buffer:=ADR(RxData); (* Memory buffer address *)
    Mdb.IFTime:=3430; (* Interframe time (uS) *)
    Mdb.Timeout:=500; (* Timeout time (mS) *)
    Mdb.Delay:=100; (* Delay time (mS) *)
  END_IF;

  (* Manage the modbus master communication. *)

  Serial(Open:=TRUE); (* Serial port management *)
  Mdb.File:=Serial.File; (* File pointer *)
  Mdb(); (* Modbus master *)
Mdb.Enable:=Serial.Opened AND NOT(Mdb.Done); (* Modbus enable *)
```

Esempi Modbus master su TCP

Ecco come si presenta lo stesso esempio di prima utilizzando una connessione TCP verso un'altro sistema SlimLine. Nel campo ***PeerAdd*** è possibile definire un URL al posto dell'indirizzo IP.

Definizione variabili

	Name	Type	Address	Array	Init value	Attribute	Description
1	RxData	UINT	Auto	[0..7]		..	Rx data buffer
2	Mdb	ModbusMaster	Auto	No		..	Modbus master FB
3	TCPClient	SysTCPClient	Auto	No		..	Client connection

Esempio FBD *(PTP114A670, FBD_ModbusMasterTCP)*