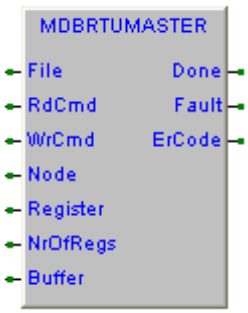
2.5 MDBRTUMASTER, Modbus Rtu master



Questo blocco funzione esegue la gestione del protocollo modbus master, è possibile definire il terminale di I/O su cui effettuare la comunicazione ***File***.

Attivando il comando di read ***RdCmd*** sul terminale di I/O viene inviato un frame modbus con il comando di ***Read Holding Registers*** (0x03) ed il valore ritornato dei registri viene trasferito nella variabile indirizzata da ***Buffer***.

Attivando il comando di write ***WrCmd*** sul terminale di I/O viene inviato un frame modbus con il comando di ***Preset Multiple Registers*** (0x10) con il valore dei registri acquisito dalla variabile indirizzata da ***Buffer***.

Terminato il comando viene attivato per un loop l'uscita ***Done***, mentre in caso di errore esecuzione comando viene attivata per un loop l'uscita ***Fault*** e viene ritornato il codice di errore in ***ErCode***.

File (FILEP) Pointer al file della risorsa come ritornato dalla funzione ***Sysfopen***.

RdCmd (BOOL) Comando di esecuzione lettura registri.

WrCmd (BOOL) Comando di esecuzione scrittura registri.

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

Register (UDINT) Indirizzo di inizio lettura o scrittura registri su nodo modbus. In accordo alle specifiche modbus il reale indirizzo inviato nel frame dati è dato da (***Register-1***) (Range da 16#0001 a 16#FFFF).

NrOfRegs (USINT) Numero di registri consecutivi da leggere o scrivere (Range da 1 a 32).

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

Done (BOOL) Attivo per un loop al termine della esecuzione del comando.

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

ErCode (USINT) Riporta codice di errore esecuzione comando (Valido se ***Fault*** e TRUE).

Esempi

Viene eseguita la lettura ogni 100 mS di 6 registri a partire da indirizzo 16#100 dal nodo modbus 1. Il valore dei registri letti è trasferito nell'array ***Registers***.

Spiando la porta seriale COM2 con un programma di emulazione terminale in grado di visualizzare i caratteri esadecimali vedremo ogni 100 mS la stringa con il comando modbus di ***Read Holding Register***: 01 03 01 00 00 06 C4 34.

Definizione variabile blocco funzione

	Name	Type	Address	Array	Init value	Attribute	Description
1	Fp	FILEP	Auto	No	0	..	Terminal I/O pointer
2	Registers	UINT	Auto	[0..9]	10(0)	..	Registers area
3	Mdb	MDBRTUMASTER	Auto	No	0	..	Modbus Rtu master FB
4	Counter	UDINT	Auto	No	0	..	Messages counter
5	Errors	UDINT	Auto	No	0	..	Errors counter
6	Sm	SYSSERIALMODE	Auto	No	0	..	Serial mode
7	AFlag	BOOL	Auto	No	FALSE	..	Auxiliary flag
8	Trigger	R_TRIG	Auto	No	0	..	Raising trigger FB

Esempio ST

```
(* ------------------------------------------------------------------- *)
(* OPEN THE COMMUNICATION PORT                                         *)
(* ------------------------------------------------------------------- *)
(* Here open the COM2 port in read/write. *)

IF (Fp = NULL) THEN
    Fp:=Sysfopen('COM2', 'rw'); (* Terminal I/O pointer *)
END_IF;

(* ------------------------------------------------------------------- *)
```

```
    (* INITIALIZATION                                                          *)
    (* ---------------------------------------------------------------------- *)
    (* Set the serial mode. *)

    IF (SysFirstLoop) THEN
        AFlag:=SysGetSerialMode(ADR(Sm), Fp); (* Get serial mode *)
        Sm.Baudrate:=57600;
        Sm.Parity:='E';
        Sm.DTRManagement:=DTR_AUTO_WO_TIMES;
         AFlag:=SysSetSerialMode(ADR(Sm), Fp); (* Set serial mode *)
    END_IF;

    (* ---------------------------------------------------------------------- *)
    (* MODBUS MASTER                                                          *)
    (* ---------------------------------------------------------------------- *)
    (* Preset the modbus master FB parameters. *)

    Mdb.File:=Fp; (* Terminal I/O pointer *)
    Mdb.Node:=1; (* Node number *)
    Mdb.Register:=257; (* Start register address *)
    Mdb.NrOfRegs:=6; (* Number of registers *)
    Mdb.Buffer:=ADR(Registers); (* Address of data buffer *)

    (* Call the modbus master FB an execute read command every 100 mS. *)

    Trigger(CLK:=SysClock100);
    Mdb(RdCmd:=Trigger.Q);

    (* Check if done or error and count them. *)

    IF (Mdb.Done) THEN Counter:=Counter+1; END_IF;
    IF (Mdb.Fault) THEN Errors:=Errors+1; END_IF;

(* [End of file] *)
```