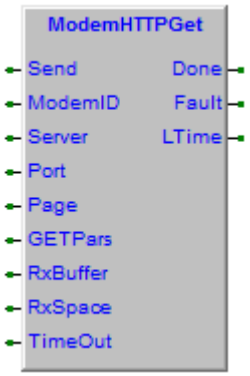
1.1.6 ModemHTTPGet, executes a HTTP Get request

Type	Library	Version
FB	eModemLib	SFR057D000

Questo blocco funzione esegue una richiesta HTTP inserendo in linea dei parametri GET, si collega al blocco funzione di gestione modem ***ModemCore***, occorre passare il ***ModemID*** in uscita dal blocco funzione di gestione modem.



Su fronte attivazione ingresso ***Send*** viene eseguita la connessione al ***Server*** HTPP sulla porta ***Port***, e richiesta la pagina ***Page***. La pagina viene richiesta con i parametri GET definiti nel buffer puntato da ***GETPars***. La pagina ricevuta dal server viene trasferita nel buffer puntato da ***RxBuffer***, i dati ricevuti che superano la dimensione del buffer ***RxSpace*** sono scartati.

Terminata la ricezione della pagina richiesta si attiva per un loop l'uscita ***Done***, se pagina non è ritornata nel tempo definito in ***TimeOut*** l'esecuzione termina con errore.

Send (BOOL) Sul fronte di attivazione comanda richiesta pagina HTTP. ***Attenzione!*** La richiesta verrà eseguita non appena il modem è libero.

ModemID (UDINT) ID modem fornito in uscita dalla ***ModemCore***.

Server (@USINT) Pointer alla stringa di definizione del server HTTP.

Port (UINT) Porta su cui effettuare la connessione.

Page (@USINT) Pointer alla stringa di definizione pagina richiesta.

GETPars (@USINT) Pointer alla stringa parametri GET da inserire nella richiesta.

RxBuffer (@USINT) Pointer al buffer di ricezione pagina richiesta.

RXSpace (UINT) Dimensione in bytes del buffer di ricezione pagina.

TimeOut (UINT) Tempo massimo attesa ricezione pagina (mS).

Done (BOOL) Attivo per un loop al termine della ricezione pagina.

Fault (BOOL) Attivo per un loop se errore ricezione pagina.

LTime (UDINT) Tempo di caricamento pagina (mS).

Codici di errore

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

10042010 ***ModemID*** non definito.

10042020 ***ModemID*** non corretto.

10042080 Errore case gestione.

10042100~8 Errore gestione richiesta pagina da server HTTP.

Esempi

L'utilizzo tipico di questa FB è di collegarsi ad un server HTTP dove sono eseguite pagine script (ASP, PHP, Phyton, ecc) richiedendo la pagina fornendo in linea alla chiamata dei parametri in GET. Lo script della pagina potrà eseguire operazioni con i dati passati in linea e ritornare dei valori come risultato. Ecco un semplice esempio che tramite una pagina script PHP sul server permette di inviare una Email.

Definizione variabili

	Name	Type	Address	Array	Init value	Attribute	Description
1	Pulse	BOOL	Auto	No	FALSE	..	Auxiliary pulse
2	Result	BOOL	Auto	No	FALSE	..	Result flag
3	ABf	INT	Auto	No	0	..	Auxiliary buffer
4	RxD	STRING	Auto	[32]		..	HTTP Rx data
5	TxD	STRING	Auto	[128]		..	HTTP Tx data
6	Mdm	ModemCore_v2	Auto	No	0	..	Modem management
7	HTTP	ModemHTTPGet	Auto	No	0	..	HTTP Get
8	Sm	SYSSERIALMODE	Auto	No	0	..	Serial mode

Esempio ST *(Ptp118b000, EmailWithHTTPGet)*

```
(* ********************************************************************** *)
(* PROGRAM "EmailWithHTTPGet"                                             *)
(* ********************************************************************** *)
(* This program sends an Email by contacting a PHP script.                *)
(* ---------------------------------------------------------------------- *)

    (* ---------------------------------------------------------------------- *)
    (* INITIALIZATION                                                         *)
    (* ---------------------------------------------------------------------- *)
    (* General initializations. *)

    IF (SysFirstLoop) THEN

        (* Opening the serial port on wich the modem is connected. *)

        Mdm.File:=Sysfopen('COM0', 'rw'); (* COM port file pointer *)

        (* Set the serial communication mode. *)

        ABf:=SysGetSerialMode(ADR(Sm), Mdm.File);
        Sm.Baudrate:=115200; (* Baudrate *)
        Sm.Parity:='N'; (* Parity *)
        Sm.DataBits:=8; (* Data bits *)
        Sm.StopBits:=1; (* Stop bits *)
        Sm.DTRManagement:=DTR_OFF; (* DTR management *)
        Sm.DTRComplement:=FALSE; (* DTR complement *)
        Sm.EchoFlush:=TRUE; (* Flush the echo caracters *)
        ABf:=SysSetSerialMode(ADR(Sm), Mdm.File);

        (* Modem parameters. Please change them according your requirements. *)

        Mdm.AutoHangUp:=TRUE; (* Auto hangup *)
        Mdm.APN:=ADR('M2Mbis.vodafone.it'); (* APN definition *)
        Mdm.DNS:=ADR('8.8.8.8'); (* DNS definition *)

        (* HTTP parameters. Please change them according your requirements. *)

        HTTP.Server:=ADR('www.MyServer.com'); (* HTTP server *)
        HTTP.Port:=80; (* HTTP port *)
        HTTP.Page:=ADR('eMailSend.php'); (* Page name *)
        HTTP.TimeOut:=5000; (* Connection timeout (mS) *)
        HTTP.GETPars:=ADR(TxD); (* GET parameters pointer *)
        HTTP.RxBuffer:=ADR(RxD); (* Rx buffer pointer *)
        HTTP.RxSpace:=SIZEOF(RxD); (* Rx buffer space *)
    END_IF;

    (* ---------------------------------------------------------------------- *)
    (* FBs MANAGEMENT                                                         *)
    (* ---------------------------------------------------------------------- *)
```

```
    (* Manage the modem core. *)

    Mdm(Enable:=TRUE); (* Modem core management *)
    HTTP.ModemID:=Mdm.ModemID; (* Modem ID *)

    (* Manage the HTTP connection. *)

    HTTP();
    HTTP.Send:=FALSE;

    (* ---------------------------------------------------------------------- *)
    (* GET THE HTTP PAGE AND SEND THE EMAIL                                   *)
    (* ---------------------------------------------------------------------- *)
    (* On digital input activation the HTTP page is requested. *)

    IF (Di00M00 <> Pulse) THEN
        Pulse:=Di00M00; (* Auxiliary pulse *)

        IF (Pulse) THEN
            ABf:=SysVarsnprintf(ADR(TxD), 128, 'To=%s', STRING_TYPE, ADR('sbertana@elsist.it'));
            ABf:=SysVarsnprintf(ADR(TxD)+LEN(TxD), 128-LEN(TxD), '$26Subject=%s', STRING_TYPE,
ADR('Subject'));
            ABf:=SysVarsnprintf(ADR(TxD)+LEN(TxD), 128-LEN(TxD), '$26Text=%s', STRING_TYPE, ADR('Text'));
             HTTP.Send:=TRUE;
        END_IF;
    END_IF;

    (* Check the HTTP request answer. *)

    IF (HTTP.Done) THEN
        IF (FIND(RxD, 'Ok') = 0) THEN Result:=FALSE; ELSE Result:=TRUE; END_IF;
    END_IF;

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

Vediamo di spiegare come funziona il programma, sul fronte di attivazione dell'ingresso ***Di00M00*** viene effettuata la connessione al server ***www.MyServer.com*** e richiesta l pagina ***eMailSend.php***. In linea alla chiamata della pagina sono posti i campi GET **To=sbertana@elsist.it&Subject=Subject&Text=Text**. Da notare che il carattere “&” è stato sostituito da “$26”.

La pagina ***eMailSend.php*** sul server contiene uno script PHP del tipo:

```
<?php
    $Headers="MIME-Version: 1.0\n";
    $Headers.="Content-type: text/html; charset=iso-8859-1\n";
    $Headers.="To: ".$_REQUEST['To']."\n";
    $Headers.="From: \n";
    mail($_REQUEST['To'], $_REQUEST['Subject'], $_REQUEST['Text'], $Headers);
    echo "Ok";
    exit();
?>
```

Come si vede lo script invia una mail tramite il comando mail i cui campi sono valorizzati dai parametri posti in GET alla richiesta. Lo statement ***$_REQUEST['To']*** è valorizzato con il testo definito in ***To=***, ***$_REQUEST['Subject']*** è valorizzato con il testo definito in ***Subject=***, e così via. Come si vede è possibille passare allo script tutti i valori che si desiderano.

Il valore di ritorno dallo script definito con lo statement echo verrà trasferito nel buffer ***RxD*** del nostro programma e quindi è possibile operare su di esso con le istruzioni di gestione stringa.