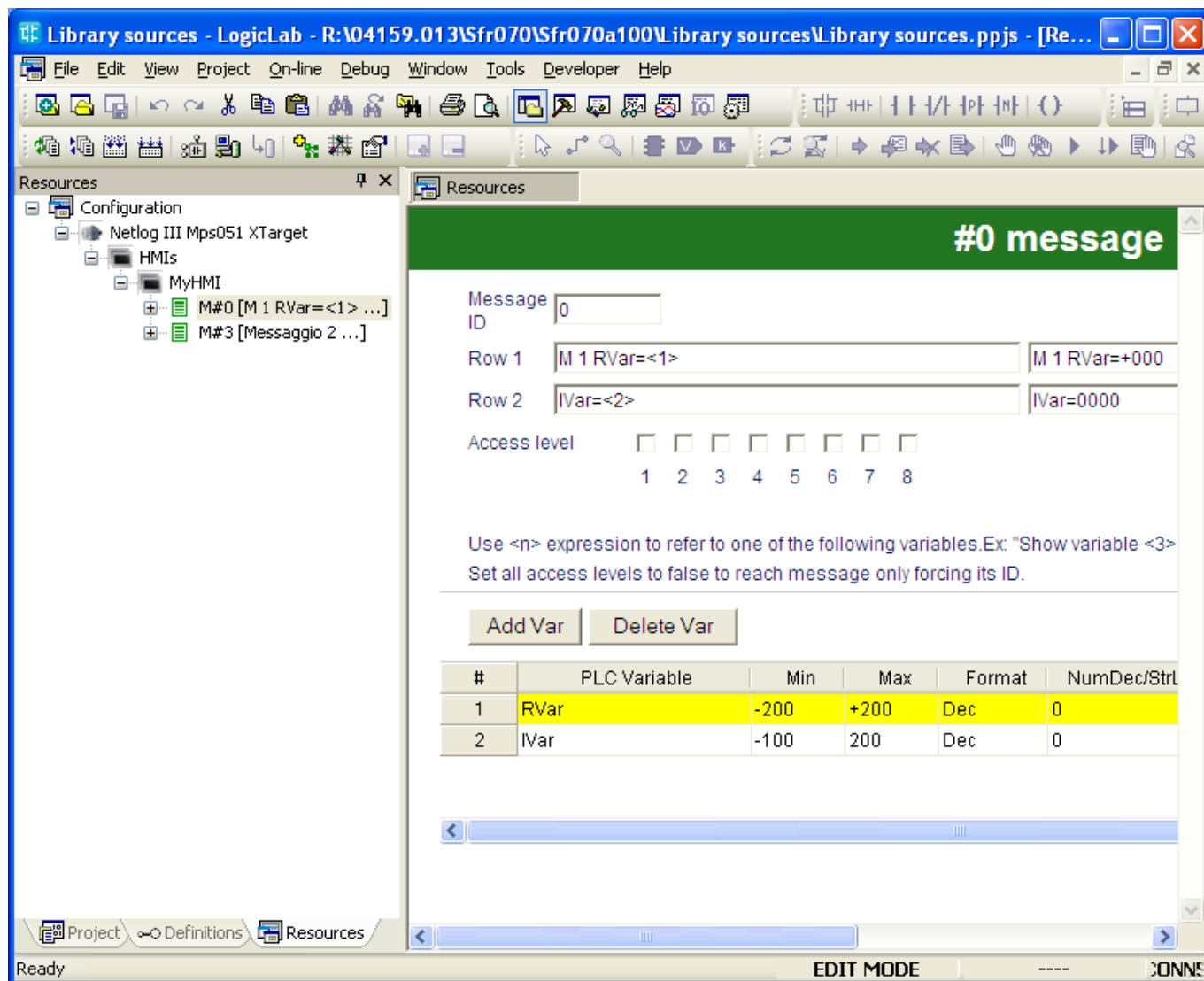


1.1 Libreria gestione interfaccia operatore (eLLabHMIBuiltInLib)

Attenzione! Per utilizzare la libreria occorre importarla nel proprio progetto. Vedere capitolo relativo all'[import delle librerie](#).

Questa libreria permette di gestire l'interfaccia operatore direttamente da LogicLab. Nel menù Resources è possibile configurare le interfacce operatore. Come si vede nella figura è stata definita una interfaccia **MyHMI** che prevede due messaggi.



Come si vede ad ogni messaggio è associato del testo ed eventuali variabili da visualizzare. Nel messaggio possono essere visualizzate tutte le variabili globali.

Nella compilazione del progetto LogicLab crea ed istanzia per ogni HMI un blocco funzione di tipo **HMIBuiltInMessages**, con il nome assegnato all'HMI, nell'esempio soprariportato **MyHMI**. Questo FB provvede alla gestione dei messaggi e delle variabili ma per la visualizzazione occorre collegare al FB una FB di gestione terminale.

1.1.1 HMIBuiltInMessages, HMI built in messages

Type	Library
FB	eLLabHMIBuiltInLib_B310

Questo blocco funzione esegue la gestione dei messaggi così come definiti da LogicLab.

Anche se ne riporto la raffigurazione a differenza degli altri blocchi funzione è automaticamente istanziato da LogicLab e non va quindi inserito dall'utente nel proprio programma. Per l'utente è importante fare riferimento alle sole variabili di ingresso ed uscita dal FB, variabili che possono essere utilizzate nel proprio programma.



Il parametro **ALevel** configura il livello di accesso ai messaggi, settando i bits della variabile si permette di visualizzare solo i messaggi che hanno lo stesso bit definito in **Access Level**.

CmdDisable (BOOL)	Disabilitazione comandi terminale. Attivandolo non è possibile eseguire operazioni.
ViewMID (UINT)	Visualizza messaggio con ID definito.
ALevel (BYTE)	Pattern livello accesso utente.
DEntryOk (BOOL)	Attivo per un loop al termine della inputazione di una variabile.
MsgID (UINT)	ID del messaggio attualmente visualizzato.
HMIBuiltInID (UDINT)	ID messaggi da passare alle FB collegate.

Codici di errore

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

- 10046010 Tabella messaggi generata da LogicLab non supportata.
- 10046020~2 Non è possibile allocare memoria per l'esecuzione.
- 10046100 Blocco funzione non eseguito in task di background (Back).
- 10046200 Messaggio con ID richiesto in **ViewMID** non trovato.
- 10046100 Spazio insufficiente per visualizzazione variabile.

1.1.2 HMIBuiltInNetlog, Netlog HMI management

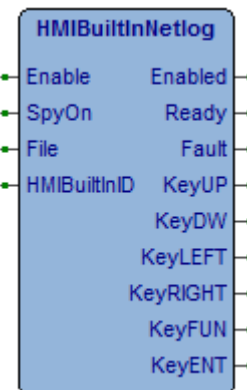
Type	Library
FB	eLLabHMIBuiltInLib_B310

Questo blocco funzione esegue la gestione del terminale (Display e tastiera) integrato nel sistema **NetlogIII**, **deve essere eseguito in task Back**. Si collega al blocco funzione **HMIBuiltInMessages** di gestione messaggi, occorre passare **HMIBuildInID** in uscita dal blocco funzione.

Attivando l'ingresso **Enable** si attiva l'uscita **Enabled** e viene gestito il terminale visualizzando i messaggi definiti nel terminale da LogicLab. Nell'ingresso **File** occorre passare lo stream del dispositivo di I/O utilizzato per la gestione (Ritornato dalla funzione **Sysfopen**).

Il FB ritorna in uscita lo stato di tutti i tasti del terminale, e questo permette di utilizzarli nel proprio programma come comandi. L'ingresso **SpyOn** se attivo permette di spiare il funzionamento della FB.

In caso di errore esecuzione viene attivata per un loop l'uscita **Fault**.



Enable (BOOL) Comando di abilitazione blocco funzione.

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

TFlags	Description
16#00000001	Rx : Ricezione dati da terminale.
16#00000002	Tx : Trasmissione comandi display.
16#00000004	Tx : Trasmissione dati display.
16#00000008	Tx : Trasmissione comando inputazione dati.

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

HMIBuiltInID (UDINT) ID gestione messaggi in uscita dalla FB (**HMIBuiltInMessages**).

Enabled (BOOL) Attivo su abilitazione blocco funzione.

Ready (BOOL) Attivo se l'hardware display è pronto.

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

KeyUP (BOOL) Stato tasto UP del terminale.

KeyDW (BOOL) Stato tasto DW del terminale.

KeyLEFT (BOOL) Stato tasto LEFT del terminale.

KeyRIGHT (BOOL) Stato tasto RIGHT del terminale.

KeyFUN (BOOL) Stato tasto FUN del terminale.

KeyENT (BOOL) Stato tasto ENT del terminale.

Codici di errore

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

10047010 **HMIBuildInID** non definito.

10047020 **HMIBuildInID** non corretto.

10047100 Blocco funzione non eseguito in task di background (Back).

Esempi

Nell'esempio è gestito il terminale di un sistema **NetlogIII**, i messaggi sono stati definiti nel programma LogicLab con il nome **MyHMI**.

Definizione variabili

	Name	Type	Address	Array	Init value	Attribute	Description
1	HMI	HMIBuiltInNetlog	Auto	No		..	Netlog terminal
2	MsgID	UINT	Auto	No		..	Displayed message ID
3	HMIPort	SysSerialPort	Auto	No		..	HMI port management

Esempio ST (PTP148A000, ST_HMIBuiltInNetlog)

```
(* Program initialization. *)

IF (SysFirstLoop) THEN

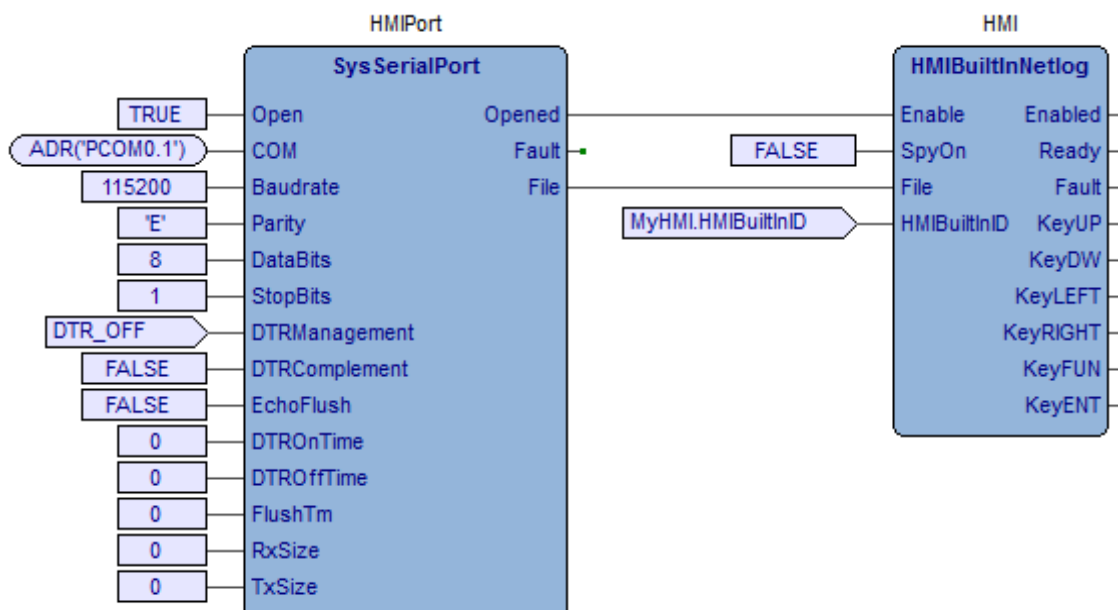
    HMIPort.Open:=TRUE; (* Open port *)
    HMIPort.COM:=ADR('PCOM0.1'); (* COM port definition *)
    HMIPort.Baudrate:=115200; (* Baudrate *)
    HMIPort.Parity:='E'; (* Parity *)
    HMIPort.DataBits:=8; (* Data bits *)
    HMIPort.StopBits:=1; (* Stop bits *)
    HMIPort.DTRManagement:=DTR_OFF; (* DTR management *)
    HMIPort.DTRComplement:=FALSE; (* DTR complement *)
    HMIPort.EchoFlush:=FALSE; (* Received echo flush *)
    HMIPort.DTROnTime:=0; (* DTR On time delay (mS) *)
    HMIPort.DTROffTime:=0; (* DTR Off time delay (mS) *)
    HMIPort.FlushTm:=0; (* Flush time (mS) *)
    HMIPort.RxSize:=0; (* Rx buffer size *)
    HMIPort.TxSize:=0; (* Tx buffer size *)

    HMI.SpyOn:=TRUE; (* Spy On *)
END_IF;

(* Here manages the Netlog terminal. *)

HMIPort(); (* HMI port management *)
HMI.File:=HMIPort.File; (* File pointer *)
HMI.HMIBuiltInID:=MyHMI.HMIBuiltInID; (* ID gestione messaggi *)
HMI(Enable:=HMIPort.Opened); (* Netlog terminal *)
```

Esempio FBD (PTP148A000, FBD_HMIBuiltInNetlog)



1.1.3 HMIPicoface, Picoface HMI management

Type	Library
FB	eLLabHMIBuiltInLib_B310

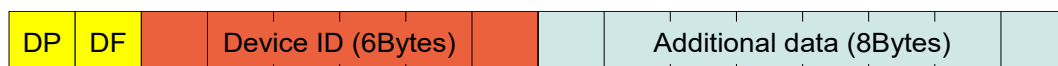
Questo blocco funzione esegue la gestione del terminale Picoface, il terminale può essere connesso al sistema su bus di estensione I2C oppure in seriale sia in configurazione FullDuplexRS232 che in configurazione HalfDuplex RS485. Il parametro **CType** definisce il tipo di comunicazione utilizzata

Il **FB deve essere eseguito in task Back**. Si collega al blocco funzione **HMIBuiltInMessages** di gestione messaggi, occorre passare **HMIBuiltInID** in uscita dal blocco funzione.

Attivando l'ingresso **Enable** si attiva l'uscita **Enabled** e viene gestito il terminale visualizzando i messaggi definiti nel terminale da LogicLab. Nell'ingresso **File** occorre passare lo stream del dispositivo di I/O utilizzato per la gestione (Ritornato dalla funzione **Sysfopen**).

Attivando i bits in ingresso è possibile comandare i LEDs e le uscite sul terminale, in uscita è ritornato lo stato dei tasti e degli ingressi del terminale. Questo permette di utilizzarli nel proprio programma come comandi.

E' prevista la gestione di un dispositivo One Wire, il FB ritorna in **OneWireData** un array del tipo.



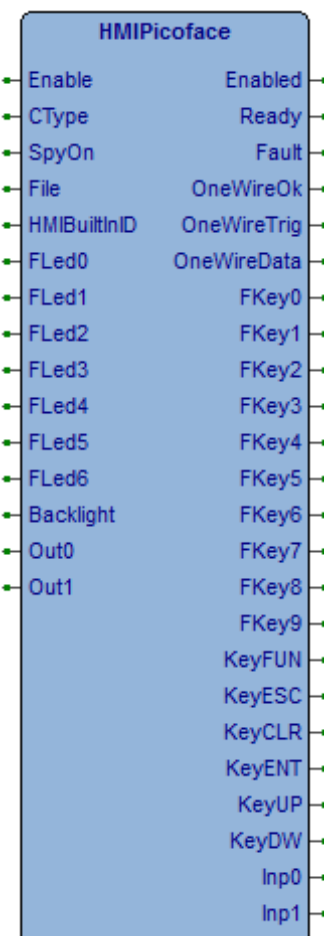
Dove:

DP: Device Protocol.

DF: Device family.

L'ingresso **SpyOn** se attivo permette di spiare il funzionamento della FB.

In caso di errore esecuzione viene attivata per un loop l'uscita **Fault**.



Comando di abilitazione blocco funzione.

Enable (BOOL)

CType (BOOL)

Tipo comunicazione. **FALSE:** Full duplex (Default), **TRUE:** Half duplex

SpyOn (BOOL)

Se attivo permette di spiare il funzionamento della FB.

TFlags Description

16#00000001 **Rx:** Ricezione frame dati da terminale.

16#00000002 **Tx:** Trasmissione frame dati verso terminale.

16#10000000 **Wn:** Messaggio di warning (Su anomalia di comunicazione).

File (FILEP)

Flusso dati **stream** ritornato dalla funzione **Sysfopen**. Per il terminale di default connesso in I2C definire **PCOM15.1**.

HMIBuiltInID (UDINT)

ID gestione messaggi in uscita dalla FB (**HMIBuiltInMessages**).

FLed0~6 (BOOL)

Comando LED tasti funzione da [0] a [6].

Backlight (BOOL)

Comando accensione backlight display.

Out0~1 (BOOL)

Comando uscite logiche.

Enabled (BOOL)

Attivo su abilitazione blocco funzione.

Ready (BOOL)

Attivo se l'hardware display è pronto.

Fault (BOOL)

Attivo per un loop su errore esecuzione.

OneWireOk (BOOL)	Attivo se almeno un dispositivo One-Wire è connesso al connettore del terminale.
OneWireTrig (BOOL)	Attivo per un loop di programma, da usare come trigger sull'array OneWireData .
OneWireData (BYTE[16])	Ritorna i dati letti dal dispositivo One Wire connesso al connettore del terminale.
FKey0~9 (BOOL)	Stato tasti funzione da [0] a [9].
KeyFUN (BOOL)	Stato tasto FUN del terminale.
KeyESC (BOOL)	Stato tasto ESC del terminale.
KeyCLR (BOOL)	Stato tasto CLR del terminale.
KeyENT (BOOL)	Stato tasto ENT del terminale.
KeyUP (BOOL)	Stato tasto UP del terminale.
KeyDOWN (BOOL)	Stato tasto DOWN del terminale.
Inp0~1 (BOOL)	Stato ingressi digitali.

Codici di errore

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

10049010 **HMIBuildInID** non definito.

10049020 **HMIBuildInID** non corretto.

10049100 Blocco funzione non eseguito in task di background (Back).

Esempi

Nell'esempio è gestito un terminale **Picoface**, i messaggi sono stati definiti nel programma LogicLab con il nome **MyHMI**.

Definizione variabili

	Name	Type	Address	Array	Init value	Attribute	Description
1	HMI	HMIPicoface	Auto	No		..	Picoface
2	HMIPort	SysSerialPort	Auto	No		..	HMI communication port

Esempio ST (PTP148A000, ST_HMIPicoface)

```
(* Program initialization. *)

IF (SysFirstLoop) THEN

    HMIPort.Open:=TRUE; (* Open port *)
    HMIPort.COM:=ADR('PCOM15.1'); (* COM port definition *)
    HMIPort.Baudrate:=115200; (* Baudrate *)
    HMIPort.Parity:='E'; (* Parity *)
    HMIPort.DataBits:=8; (* Data bits *)
    HMIPort.StopBits:=1; (* Stop bits *)
    HMIPort.DTRManagement:=DTR_OFF; (* DTR management *)
    HMIPort.DTRComplement:=FALSE; (* DTR complement *)
    HMIPort.EchoFlush:=FALSE; (* Received echo flush *)
    HMIPort.DTROnTime:=0; (* DTR On time delay (mS) *)
    HMIPort.DTROffTime:=0; (* DTR Off time delay (mS) *)
    HMIPort.FlushTm:=0; (* Flush time (mS) *)
    HMIPort.RxSize:=0; (* Rx buffer size *)
    HMIPort.TxSize:=0; (* Tx buffer size *)

    HMI.SpyOn:=TRUE; (* Spy On *)
    HMI.CType:=FALSE; (* Communication type *)
END_IF;

(* Here manages the Picoface terminal. *)

HMIPort(); (* HMI port management *)
HMI.File:=HMIPort.File; (* File pointer *)
HMI.HMIBuiltInID:=MyHMI.HMIBuiltInID; (* ID gestione messaggi *)
HMI(Enable:=HMIPort.Opened); (* Picoface terminal *)
```

Esempio FBD (PTP148A000, FBD_HMIPicoface)

