

Some skill to operate RS-232 / 422 / 485 serial COM Port by COM functions

ICP DAS ISaGRAF controllers support below Serial COM Port (RS-232/422/485) protocols:

Modbus RTU Slave	Refer to Chapter 4 of the ISaGRAF user's manual & respective getting started manual
I-7000 and I-87xxx RS-485 I/O	Refer to Chapter 6 of the ISaGRAF user's manual
Modbus RTU Master (M-7000)	Refer to Chapter 8 and 21 of the ISaGRAF user's manual
Modbus ASCII Master	Refer to Chapter 8 of the ISaGRAF user's manual
Modem Link	Refer to Chapter 13 of the ISaGRAF user's manual
MMICON	Refer to Chapter 16 of the ISaGRAF user's manual
SMS : Short Message Service	Refer to Chapter 17 of the ISaGRAF user's manual

User can apply below COM functions to operate other user-defined protocols or 3rd party protocols. (Please refer to Appendix A.4 of the ISaGRAF user's manual for description of these COM functions)

COMOPEN	Open Serial COM Port (without "Flow control" parameter)
COMOPEN2	Open Serial COM Port (with "Flow control" parameter, not for I-8xx7)
COMREADY	Test if any byte come in
COMARY_R	Read all bytes which already come in to a byte array
COMARY_W	Write many bytes in a byte array to COM Port
COMREAD	Read one bytes (Please call "COMREADY" to test first, if there is data, then "COMREAD" can be called)
COMCLEAR	Clear all received bytes in the receiving buffer
COMARY_NW	Write one signed long Integer to COM Port, format is Binary, 4-byte
COMARY_WW	Write one signed Word to COM Port, format is Binary, 2-byte
COMSTR_W	Write one string to COM Port
COMWRITE	Write one byte to COM Port
COMCLOSE	Close Serial COM Port

Note:

1. The default shipping of I-8xx7 controller has set its COM1 and COM2 (COM2:RS-485 is only for I-8417/8817) as Modbus RTU Slave Port. User can choose to switch off the COM1:Modbus RTU Slave function to become a freely used COM port by the above listed COM functions. (Please refer to Appendix C.1). To use I-8xx7 's COM5 to COM20, Please refer to Chapter 1.8 to install I-8112/8114/8142/8144 serial expansion boards.

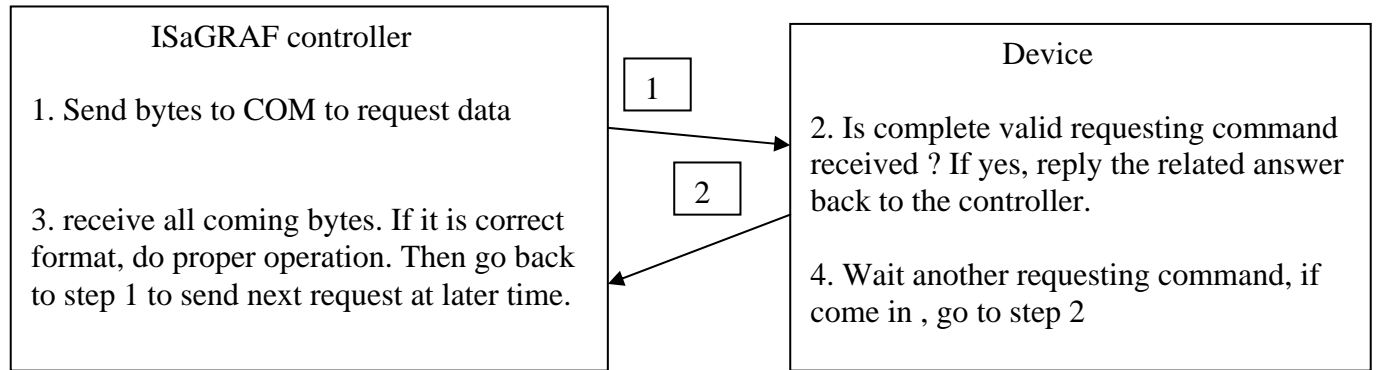
2. Wincon-8xx7 / 8xx6 's COM2 / COM3 can be switched ON as a Modbus RTU Slave Port. Or Switch Off for freely used. (Please refer to Appendix A.2 of its Getting Started manual delivered with the hardware). To use Wincon 's COM5 to COM14 at I-8112/8114/8142/8144 serial expansion boards, please refer to Appendix E of the "Getting Started:Wincon ISaGRAF PAC" manual.

3. COM1 of I-7188EG / 7186EG is set as Modbus RTU Slave port when shipping. User may switch it OFF to freely use it by COM port functions. (Please refer to its "Getting Started Manual" delivered with its hardware). However I-7188XG 's COM1 can not be switch OFF, it is always Modbus RTU Slave port. If user want to use COM3 to COM8 of I-7188EG/XG and 7186EG, please plug one extra X-5xx expansion I/O board inside it .

http://www.icpdas.com/products/PAC/i-o_expansion/x_list.htm

1. Controller send one request and then get one reply from device

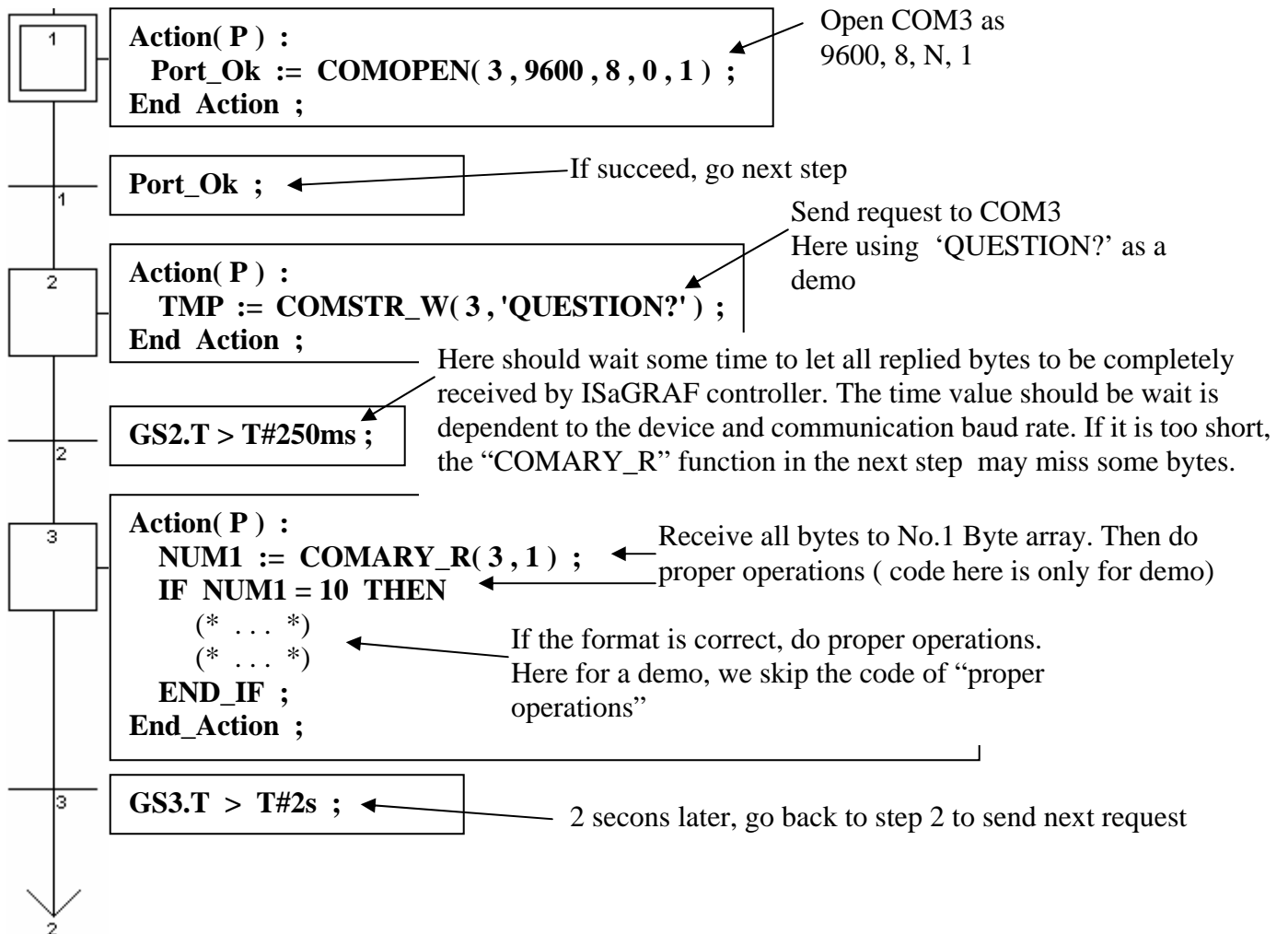
Below figure lists the most common RS-232 / 422 / 485 application.



User can use the below code or similar code to do it.

Below example will send a string "QUESTION?" to device via COM3, then waiting device to reply the related answer. And then 2 seconds later, send next same question to device, ...

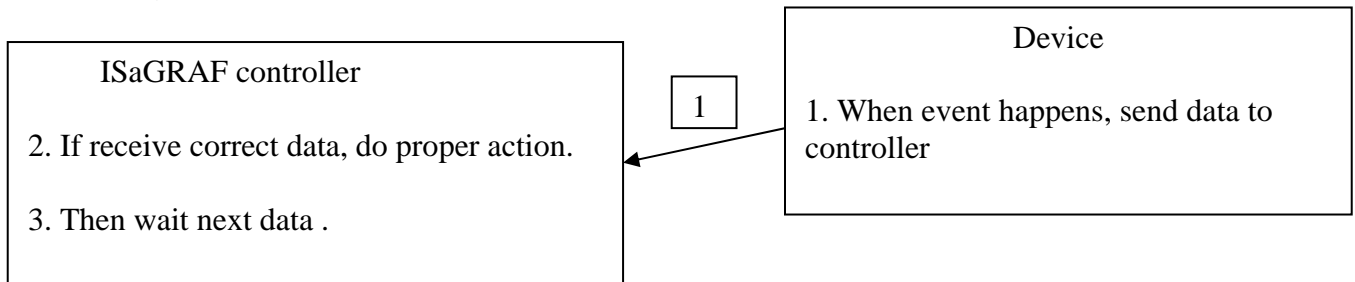
SFC program: ("Port_OK" & "TMP" is Boolean Internal, "NUM1" is Integer Internal)



2. Controller just wait data from the remote device

This kind of application is very common in the store. Like the device of “Bar code reader”, when it reads bar code on the product, it will send the related data to the controller via RS-232 / 422 / 485. The controller just receive it, not necessary to send any byte to device.

(Demo program and more information at www.icpdas.com – FAQ – Software - ISaGRAF - 066)



ST program:

```

IF INIT THEN
  INIT := FALSE ;
  Port_Ok := COMOPEN( 3 , 9600 , 8 , 0 , 1 ) ;
  T1 := T#0s ;
  STEP := 0 ;
END_IF ;
  
```

Variable declaration

“INIT” is Boolean Internal, init as TRUE
 “Port_Ok” & “TMP” is Boolean Internal
 “T1” is Timer Internal
 “STEP” & “NUM1” is Integer Internal

Open COM3 as 9600, 8, N,1

```

IF Port_Ok = False THEN
  Return ;
END_IF ;
  
```

If open fail, just quit this ST program

CASE STEP OF

```

0 : IF COMREADY(3) THEN
  STEP := 1 ;
  T1 := T#0s ;
  Tstart(T1) ;
END_IF ;
  
```

STEP=0 means “waiting state”, it should check if any byte come in. If “COMREADY” returns TRUE, it means there is at least one byte come in. Then set “STEP” to 1 and start timer “T1” to tick.

```

1 : IF T1 > T#250ms THEN
  Tstop(T1) ;
  T1 := T#0s ;
  STEP := 0 ;
  NUM1 := COMARY_R(3, 1) ;
  
```

STEP=1 means “data is coming”, when “T1” tick to the specified waiting time, call “COMARY_R” to receive all bytes to No.1 byte array (this will make sure all bytes from the “device” all well received). This waiting time is dependent with the “Device” and the communication baud rate. Here we use 0.25 second as a demo. (If setting too short than your device should have, some bytes may be missing) Remember to set “STEP” to 0 to wait next data.

```

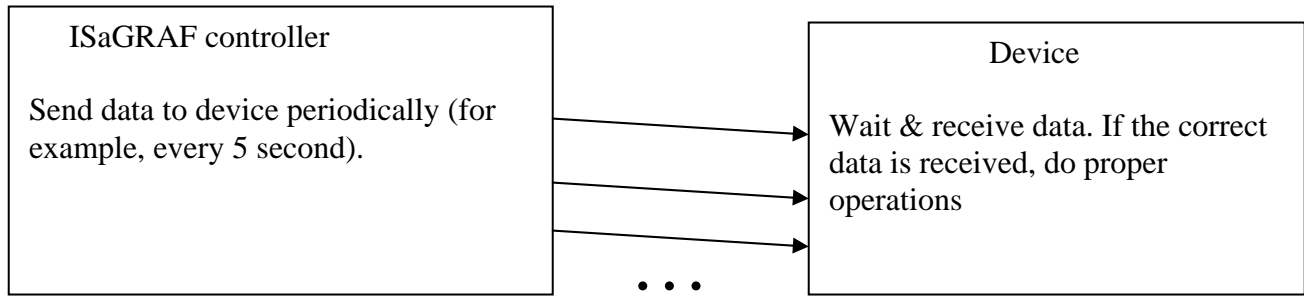
  IF NUM1=10 THEN
    (* ... *)
  END_IF ;
END_IF ;
  
```

Do proper operation if receive correct data.
 The code listed here is only for demo. (It is depend with your application, we just assume the correct data has 10 bytes here.)

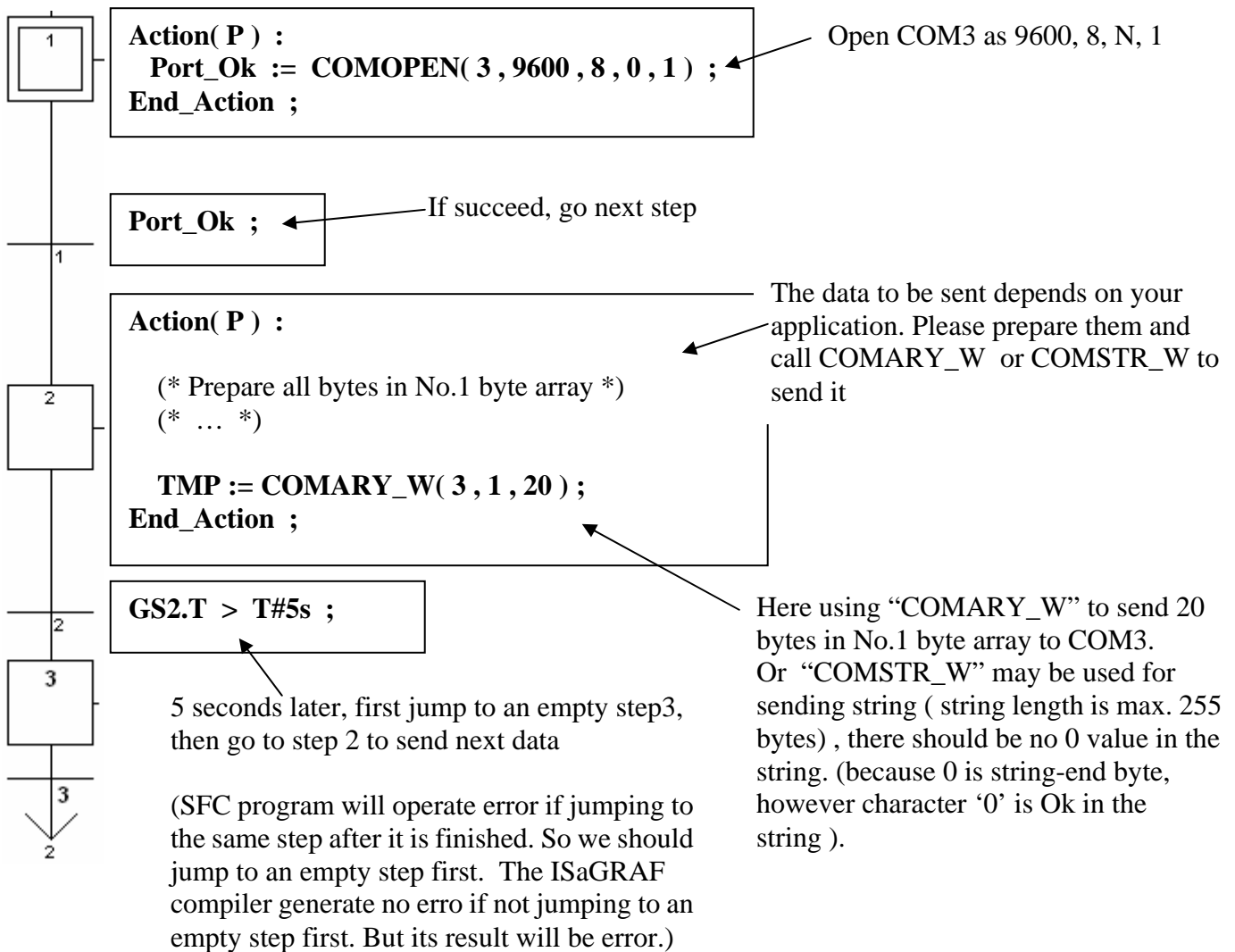
```

END_CASE ;
  
```

3. Report data to remote device periodically



SFC program: (Please declare "TMP" & "Port_Ok" as Boolean Internal)



4. Controller send data to COM port when event happens

This demo program can be running in Wincon-8xx7/8xx6 or in I-8xx7. Please init "PORT" as 2 if your target is Wincon, while 3 for I-8xx7.

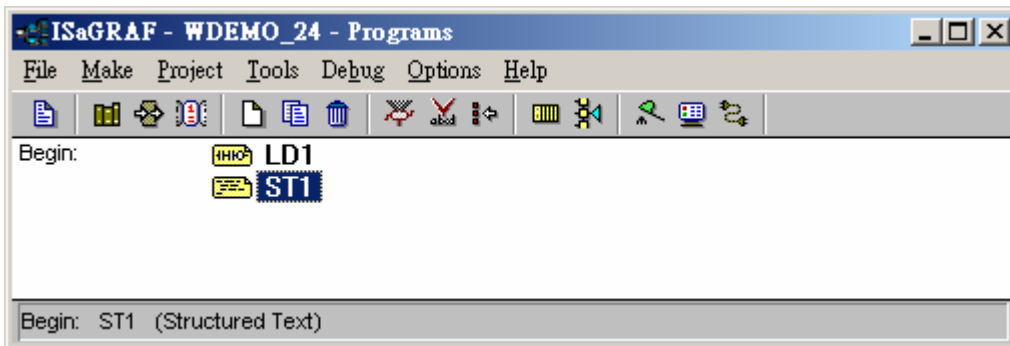
This example program is at W-8xx7 CD-ROM:\napdos\isagraf\wincon\demo\ "wdemo_24" or i-8xx7's CD-ROM:\napdos\isagraf\8000\demo\ "demo_70" or ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/ "wdemo_24" or www.icpdas.com – FAQ – Software – ISaGRSAF – FAQ059

We use "Variable array" in this example program, please refer to Chapter 2.6 (or FAQ039) for detailed description of "Variable array"

Variables :

Name	Type	Attribute	Description
INIT	Boolean	Internal	Init as TRUE, True indicates first PLC scan cycle
TMP	Boolean	Internal	Temporary using
Tick1	Boolean	Internal	pulse generated every 1 sec to counting time
IN[0..7]	Boolean	Input	input of ch1 to 8 at slot 1: 8077 , Variable array, Dim=8
Old_IN[0..7]	Boolean	Internal	Old value of IN[0..7] , Variable array, Dim=8
ii	Integer	Internal	Index of "For" loops
Port	Integer	Internal	COM PORT Number to open, init as 2 for Wincon
CNT[0..7]	Integer	Internal	time of True state of IN[0..7], Variable array, Dim=8, unit is sec
Msg1	Message	Internal	Message to send to COM2, init length as 128

Project architecture:



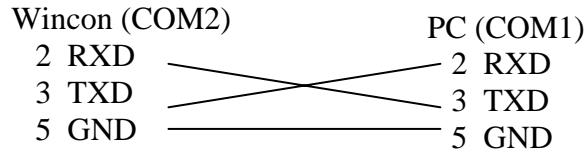
Operations:

1. If IN[0..7] rising from False to True and hold in True for at least 3 sec, send one message = 'Alarm N' + <LF> <CR> to COM2. N= 1,2, ... 8 depends on which Input is triggered. For ex, if IN[2] is rising and hold in True longer than 3 seconds, send 'Alarm 3' + <LF> <CR> to COM2
2. If after IN[0..7] 's first alarm is sent and then continusly hold in True for 30 seconds, then send one more messge to COM2 after every 30 second past until the state of IN[0..7] is falling to FALSE. The string is for ex, 'Alarm 3 , 30 sec past !'

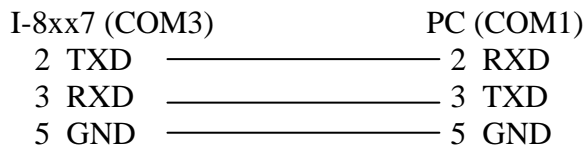
How to test ?

1. Please download wdemo_24 to W-8xx7+ slot 1: I-8077 (or demo_70 for I-8xx7+slot 0: I-8077)

2. Connect a RS232 cable between W-8xx7's COM2 to your PC's COM1

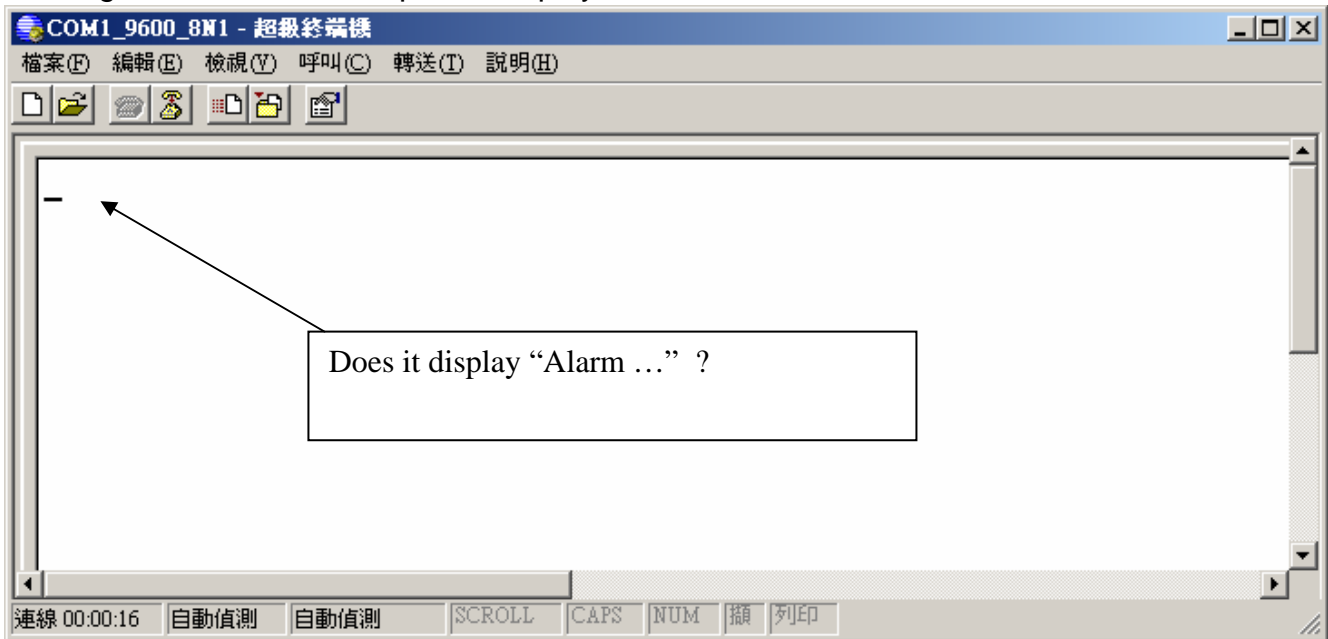


Or if you are using I-8xx7's COM3 to your PC's COM1



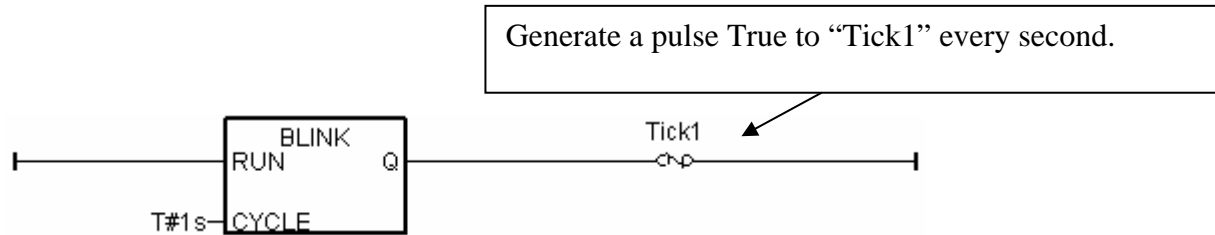
3. Open PC's Hyper terminal at COM1 with 9600, 8 char. size, no parity, 1 stop bit and No flow control. And then please switch I-8077's Input1 or 2 or ... from FALSE to TRUE and wait about three seconds. If it works, there should be a message "Alarm ..."

displayed. And then please hold this input TRUE more than 30 seconds, there should be one another message "Alarm ..., 30 sec past !" displayed.



Program description:

LD1 program



ST1 program:

If INIT then

INIT := FALSE ; (* only do it in 1st PLC scan *)

TMP := COMOPEN(PORT , 9600 , 8 , 0 , 1) ; (* Open COM2 as 9600,8,N,1 *)

(* init CNT[0..7] as -7 *)

for ii := 0 to 7 do

CNT[ii] := -7 ;

end_for ;

end_if ;

for ii := 0 to 7 do

(* test if IN[0..7] rising from False to True *)

if (IN[ii] = True) and (OLD_IN[ii] = False) then

(* Input been triggered, set related CNT[] value as -3 *)

(* if CNT[] value is not -7 , it means "the related input is triggered" *)

(* Then CNT[] will increase by 1 every second unless the related input reset to False *)

CNT[ii] := -3 ;

end_if ;

(* if Input is reset to False, set related CNT[] value as -7: "Not triggered" *)

if IN[ii] = False then

(* reset related CNT[] value to -7: "Not triggered" *)

CNT[ii] := -7 ;

end_if ;

if Tick1 then (* Tick1 is generated as a pulse True every second in "LD1" program *)

(* if CNT[] is larger than -7 , it means "the related input is triggered" *)

if CNT[ii] > -7 then

```
CNT[ii] := CNT[ii] + 1 ; (* CNT[ ] plus 1 , if Tick1= True, it means one second past *)
```

```
(* ----- *)  
(* if the related input is triggered and hold in True more than 3 second *)  
if ( CNT[ii] = 0 ) then (* if CNT[ ] is from -3, -2, -1 to 0 , send first Alarm message *)  
  CNT[ii] := 0 ; (* reset as 0 , prepare to tick to 30 second *)  
  (* Send message to COM2 *)  
  msg1 := 'Alarm ' + MSG( ii + 1 ) + '$0A$0D' ;  
  TMP := comstr_w( PORT , msg1 ) ;  
end_if ;  
(* ----- *)
```

```
(* ----- *)  
(* if the triggered input hold in True more than 30 seconds, send next message to COM2 *)  
if ( CNT[ii] = 30 ) then (*if CNT[ ] is from 1, 2, ... to 30 , send next message *)  
  CNT[ii] := 0 ; (* reset as 0 , prepare to tick to 30 second *)  
  (*Send message to COM2 *)  
  msg1 := 'Alarm ' + MSG( ii + 1 ) + ', 30 sec past ! $0A$0D' ;  
  TMP := comstr_w( PORT , msg1 ) ;  
end_if ;  
(* ----- *)
```

```
end_if ; (* "if CNT[ ] > -7 then" *)
```

```
end_if ; (* "if Tick1 then" *)
```

```
(* update value of OLD_IN[ ] *)  
OLD_IN[ii] := IN[ii] ;
```

```
end_for ;
```