Frequently Asked Questions

netSCRIPT

Tips and Tricks

Hilscher Gesellschaft für Systemautomation mbH

www.hilscher.com

DOC100208FAQ02EN | Revision 2 | English | 2010-03 | Released | Public

Table of Contents

1 Introduction

1.1 Introduction About this Manual

This manual provides you the frequently asked questions and their answers all about netSCRIPT.

1.1.1 List of Revisions

Index	Date	Chapter	Revision
1	2010-02-10	All	Created
2	2010-03-12	2.2	Performance statements included
			Functions with multiple return values
			Answer of question about `return` command extended
		2.3	Explanation of behavior in case of fatal errors
		2.4	Included default cycle time of the debugger

Table 1: List of Revisions

1.1.2 Conventions in this Manual

Operation instructions, a result of an operation step or notes are marked as follows:

Operation Instructions:

➢ <instruction>

Or

1. <instruction>
2. <instruction>

Results:

⇳ <result>

Notes:

Important: <important note>

Note: <note>

<note, were to find further information>

1.2 Legal Notes

1.2.1 Copyright

1.2.2 Important Notes

The user manual, accompanying texts and the documentation were created for the use of the products by qualified experts, however, errors cannot be ruled out. For this reason, no guarantee can be made and neither juristic responsibility for erroneous information nor any liability can be assumed. Descriptions, accompanying texts and documentation included in the user manual do not present a guarantee nor any information about proper use as stipulated in the contract or a warranted feature. It cannot be ruled out that the user manual, the accompanying texts and the documentation do not correspond exactly to the described features, standards or other data of the delivered product. No warranty or guarantee regarding the correctness or accuracy of the information is assumed.

We reserve the right to change our products and their specification as well as related user manuals, accompanying texts and documentation at all times and without advance notice, without obligation to report the change. Changes will be included in future manuals and do not constitute any obligations. There is no entitlement to revisions of delivered documents. The manual delivered with the product applies.

Hilscher Gesellschaft für Systemautomation mbH is not liable under any circumstances for direct, indirect, incidental or follow-on damage or loss of earnings resulting from the use of the information contained in this publication.

1.2.3 Exclusion of Liability

The software was produced and tested with utmost care by Hilscher Gesellschaft für Systemautomation mbH and is made available as is. No warranty can be assumed for the performance and flawlessness of the software for all usage conditions and cases and for the results produced when utilized by the user. Liability for any damages that may result from the use of the hardware or software or related documents, is limited to cases of intent or grossly negligent violation of significant contractual obligations. Indemnity claims for the violation of significant contractual obligations are limited to damages that are foreseeable and typical for this type of contract.

It is strictly prohibited to use the software in the following areas:

- for military purposes or in weapon systems;
- for the design, construction, maintenance or operation of nuclear facilities;
- in air traffic control systems, air traffic or air traffic communication systems;
- in life support systems;
- in systems in which failures in the software could lead to personal injury or injuries leading to death.

We inform you that the software was not developed for use in dangerous environments requiring fail-proof control mechanisms. Use of the software in such an environment occurs at your own risk. No liability is assumed for damages or losses due to unauthorized use.

1.2.4 Warranty

Although the hardware and software was developed with utmost care and tested intensively, Hilscher Gesellschaft für Systemautomation mbH does not guarantee its suitability for any purpose not confirmed in writing. It cannot be guaranteed that the hardware and software will meet your requirements, that the use of the software operates without interruption and that the software is free of errors. No guarantee is made regarding infringements, violations of patents, rights of ownership or the freedom from interference by third parties. No additional guarantees or assurances are made regarding marketability, freedom of defect of title, integration or usability for certain purposes unless they are required in accordance with the law and cannot be limited. Warranty claims are limited to the right to claim rectification.

1.2.5 Export Regulations

The delivered product (including the technical data) is subject to export or import laws as well as the associated regulations of different counters, in particular those of Germany and the USA. The software may not be exported to countries where this is prohibited by the United States Export Administration Act and its additional provisions. You are obligated to comply with the regulations at your personal responsibility. We wish to inform you that you may require permission from state authorities to export, re-export or import the product.

1.2.6 Registered Trademarks

Windows® 2000/Windows® XP are registered trademarks of Microsoft Corporation.

All other mentioned trademarks are property of their respective legal owners.

2 Frequently asked Questions

2.1 Introduction - netSCRIPT Achievements within Minutes

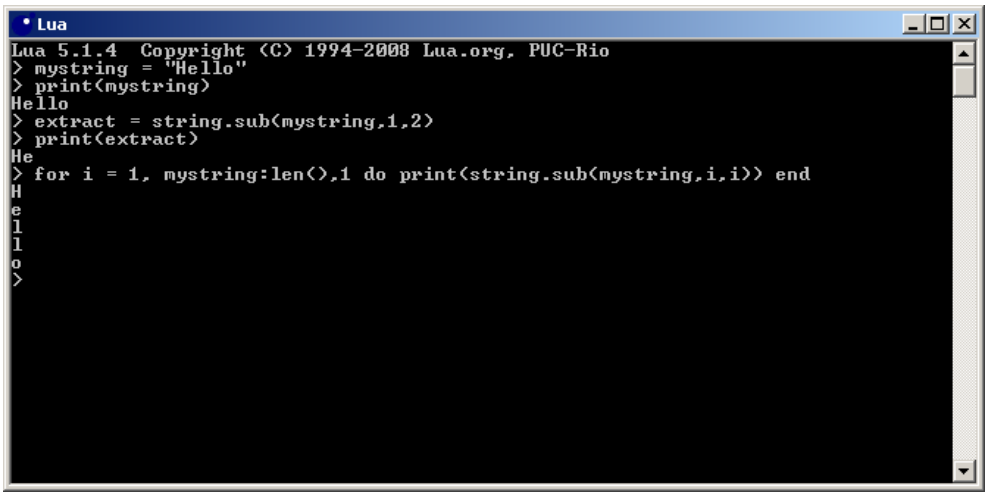
Within minutes the following steps are leading to your first netSCRIPT achievement. The whole procedure can be viewed as video tutorial on the device's DVD-ROM.

1. Get a gateway NT 100-XX-RS with serial interface
2. Connect the gateway with your PC via the USB configuration port
3. Connect the serial port of your gateway to an available COM port of your PC
4. Start the Windows HyperTerminal program on this PC and configure the serial port to 115200, 8, N, 1.
5. Install the configuration tool SYCON.net and start it
6. Choose the correct gateway from the device catalogue and download the corresponding firmware to it
7. Install the netSCRIPT debugger and start it
8. Load in the „Hello World“ example and load it to the gateway and start it afterwards
9. Check the reception of the text in the HyperTerminal program

2.2 Questions to the Language Specifics

Can netSCRIPT respectively Lua be tested under Windows?

Yes you can download a Lua interpreter at http://luaforwindows.luaforge.net/ and test Lua functionality under Windows without any additional hardware.



Which differences between netSCRIPT and „C“ are most noticeable?

There is no need for a data type when a variable is defined. There are no header files and there doesn't exist the `#define` command. All definitions are always set up as variables

How can a Variable be created?

All strings in the source code that are different from the keywords of the language itself are interpreted as names. For example `whereIam` or `I_AM_HERE` are just names in the first instance. Only an assignment transforms a simple name into a variable. `whereIam = "I am here"` is creating a variable `whereIam` of a data type `string`.

How are Variables internally managed?

Variables are internally held in a table. Within each entry a variable has name and a value. Only the value decides on the data type. If there doesn't exist a variable in the table, it has the value `nil`.

Do variables need a data type?

If an assignment do create a variable, the variable gets implicitly the data type based on the value that has been assigned. It is also possible that a variable is changing its data type with the next assignment.

Example code:

```
mystring = “Hello“  -- is of type string
mynumber = 5 -- is of type number
dynvar = 5  -- is of type number now
dynvar = “Hello“ -- is of type string now
```

Are there keywords that are not allowed to be used as names?

Yes. There are altogether 21 keywords reserved that cannot be used as names, such as `if`, `then`, `else`, `and` etc.

Is netSCRIPT case-sensitive?

Yes. A `hello` or `Hello` are two different names.

Do you need to pass all parameters of a function always?

No. Expected parameters that are not passed in a function call are automatically set to value `nil` when the function is called.

How can you return from a function before its end?

With the `return` command. If the command is used in the rootbranch of the script, the script execution will be terminated and cut short. After the next exceeded cycle time the system will jump into the script to the beginning.

May a function return multiple return values?

Yes this is possible. The example code shows the function `PortIsExchangeDone()` returning three values at a time.

Example code:

```
status, RXDATA, rxerror = MyPort:PortIsExchangeDone()

if status == port.STA_CHAR_TIMEOUT and RXDATA and not rxerror then
   -- reception successful
end
```

Are there statements to the performance of individual commands?

Yes, the following table gives an overview:

Command	Time
Number assigned to variable `'a = 5'` :	1.56 usec/exec
Floating assigned to variable `'a = 3.1416'`	1.64 usec/exec
String assigned to variable `'a = ”Hello World”'`	1.68 usec/exec
Adding `'a = 2 + 3'`	1.62 usec/exec
Multiplying `'a = 2 * 3'`	1.56 usec/exec
Dividing `'a = 2 / 3'`	1.56 usec/exec
Simple if `'if a == nil then end'`	1.72 usec/exec
Call a simple function `util.SetLed('run', true)`	12.4 usec/exec
Call simple function via object `b:BusIOSetRun(false)`	9.6 usec/exec
Format a string `'string.format('%02d:%02d:%02d',11,22,33)'`	41.7 usec/exec
Parse string `'string.match('10:30:00','^([0-2][0-9]):?([0-5][0-9]):?([0-5][0-9])?$')'`	23.88 usec/exec

Table 2: Performance

2.3 Questions to the general Program Structure

Is the script always executed periodically?

Yes the script will be executed from the very first source code line to the last one periodically.

Is it possible to program loops that are exceeding the usual cycle time of the script?

Yes. The number of missed cycles are counted internally and are catched up later when the script is following its usual process. In this case the script is called as fast as possible without observing the cycle time. Are all missed cycles catched up the script is called again in the configure cycle time.

How can you avoid that the whole script code is executed?

The script has to be divided into phases and code segments.

Which phases should be distinguished in the script?

Two phases.

A.) The initialization phase should contain those parts of the code that shall be executed just only once.

B.) The runtime phase should contain the part of the code that shall be executed periodically.

How do you realize the program init phase?

The phases should be distinguished via a state variable. The check for the value `nil` of a variable recognizes the device's cold start, since it has no value assigned.

Example code:

```
if mystate == nil then
  -- script is in init state
end
```

How do you realize additional phases or step sequences in the script?

By assigning different values to a single state variable and a conditional check for those values.

Example code:

```
if mystate == nil then
  -- script is in init state
  MY_STATE_INIT_DONE, MY_STATE_1, MY_STATE_2 = 1,2,3
  …
  mystate = MY_STATE_INIT_DONE

-- script is in running state
elseif mystate == MY_STATE_INIT_DONE then
    …
  mystate = MY_STATE_1
elseif mystate == MY_STATE_1 then
    …
  mystate = MY_STATE_1
elseif mystate == MY_STATE_2 then
    …
  mystate = MY_STATE_INIT_DONE
end
```

Do there exits constants?

netSCRIPT as such doesn't know constants, but value assignments to variables only. An assignment has to be executed once in order to let the variable get the value.

Where are constants defined?

Commands to specify constant values like `MY_STATE_INIT_DONE = 1` should be placed in the initialization phase code segment, since they need to be executed only once.

Where have functions to be defined?

A function can be defined at any place in the script. The definition must have been executed once before the function can be called, else you get a runtime error.

Which value do undefined variables have?

Since variables are managed in tables internally, a variable without an assigned value does not have an entry in the table and therefore the value `nil`. By using the assignment `myvariable = nil` an existing variable can be removed from the table again.

What else shall be programmed in the initialization phase?

A.) Assignments of constant values like `MY_STATE_INIT_DONE = 1` that need to be executed only once. This should cover also the definition of functions.

B.) Call of initialization functions of netSCRIPT specific services such as `PortOpen()` that need to be called only once during execution.

Example code:

```
if mystate == nil then
  MY_STATE_INIT_DONE, MY_STATE_1, MY_STATE_2 = 1,2,3 -- constants
  myfunction print() –- functions declarations
   ..
  end

  port = PortOpen()  -- call of initializing functions
  myio = BusIoOpen()
  …
  mystate = MY_STATE_INIT_DONE
else
```

Some functions return back errors in the variable `lasterror`, is a check necessary?

For a safe communication it is recommended to evaluate the variable `lasterror`. It is recommended as well to pass such errors to the superordinated network via the both error registers. This enables error recognition in the PLC as well and points out the origin of the error.

Is there an additional custom timer available for programming?

No. The only beat you have is the cycle time the script is called up periodically. Based on a global variable that counts the cycles you can derive relative times by subtraction.

Example code:

```
mytime = mytime + 1
…
if mystate == START then
  timesave = mytime
  mystate = WAIT_TIMEOUT
elseif mystate == WAIT_TIMEOUT
  timediff = mytime - timesave
  if timediff > 1000 then

  end
end
```

Which priority does netSCRIPT have in relation to the rest of the software components in the gateway?

netSCRIPT does have a low priority in the system. Highest priority has always the handling of the superordinated IO network protocol. This is why netSCRIPT is neither capable to disturb the IO protocol nor to stop it.

How to behave in case of fatal errors?

You should use the `assert()` function. In case the value checked is `nil` or `false` the script will be stopped and a passed error text will be deposited to be read from netSCRIPT debugger or SYCON.net.

Example code:

```
myPort = PortOpen()
assert(myPort, ”Opening the port failed”)
```

Output in netSCRIPT debugger:

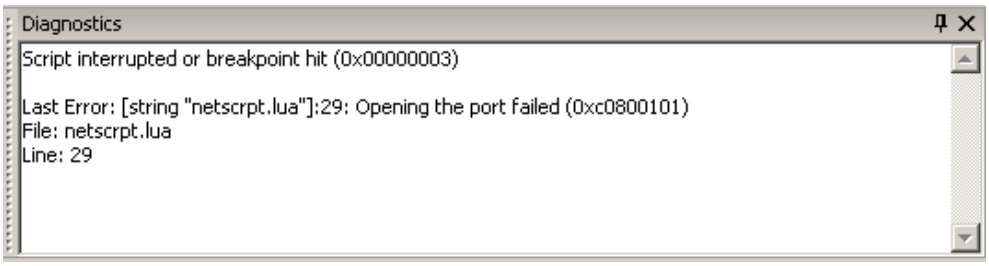


Figure 1: Output in netSCRIPT Debugger in Case of fatal Errors

Output in SYCON.net

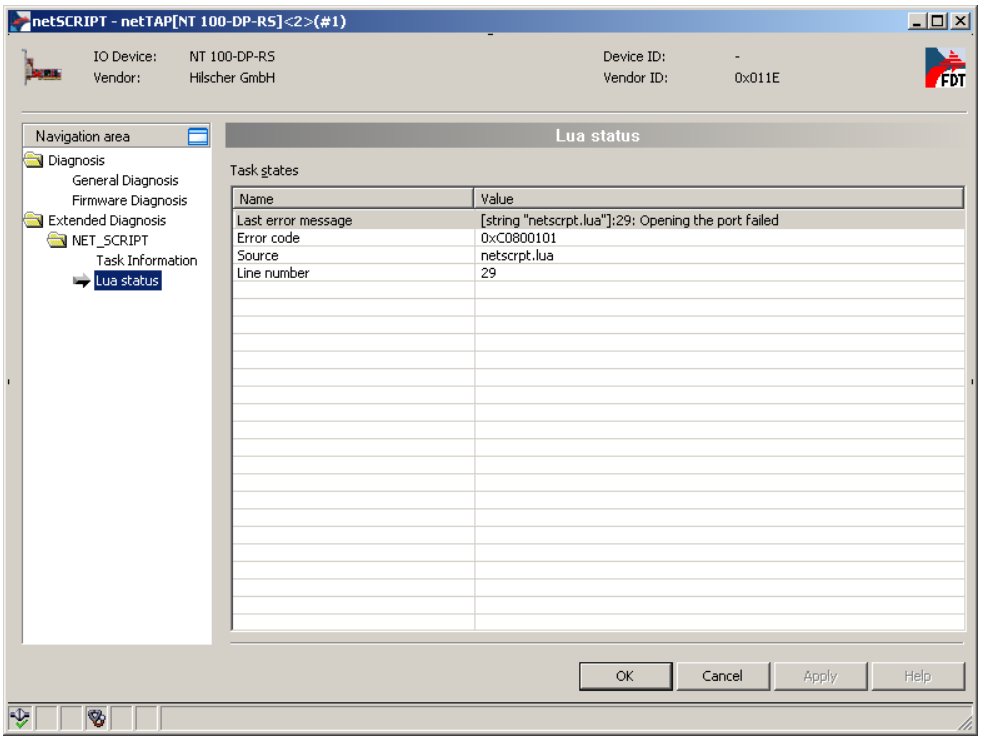


Figure 2: Output in SYCON.net in Case of fatal Errors

2.4 Question to SYCON.net and netSCRIPT Debugger

Why are there existing two utilities to load a script?

SYCON.net and netSCRIPT debugger are those two tools for netSCRIPT dealings. SYCON.net configures the whole gateway. It loads the script file and the parameter of the superordinated network. The netSCRIPT debugger is to be used just temporarily for debugging purposes until the script is error free. The final script has to be download with SYCON.net at the end.

Which is the order to use both tools?

You have to begin with SYCON.net first in order to configure the whole gateway. This enables the IO communication to the superordinated network. During this initial configuration there is no need to specify a script file. Afterwards you have to use the netSCRIPT debugger to create and debug the script file until it is running. The final script file has to be downloaded via SYCON.net.

Is the script file created with SYCON.net or netSCRIPT debugger?

SYCON.net has a script file editor, a syntax checker and can download a script file with the corresponding IO network configuration. But SYCON.net has no debug capabilities. This is why SYCON.net should be used to download ready and error free script files only.

Are SYCON.net and netSCRIPT debugger able to access the gateway via USB simultaneously?

Yes. It is possible to connect both applications the same time to the gateway.

Do I have to use netSCRIPT debugger as skilled programmer?

No, if you are skilled enough you may use SYCON.net only and go without netSCRIPT debugger.

Where is the script cycle time configured?

You can configure the cycle time in SYCON.net only. The netSCRIPT debugger does not allow to configure it and uses a default value of 10msec. In case a script file shall get a cycle time in it final release state when it is error free it has to be downloaded with SYCON.net in the last instance to configure it with the right cycle time.

Is the script that has been downloaded via netSCRIPT saved persistent in the gateway?

Yes. The netSCRIPT debugger uses the gateway's file system and stores the script file permanently.

Does the access via netSCRIPT debugger work even if SYCON.net has not preconfigured the gateway?

Yes. With netSCRIPT debugger you can get access at any time. In case there is no valid configuration stored the parameters for the superordinated network are missing and hence the gateway can't get operative via the IO network.

What is the difference between the commands „Sync" and „Reload" in netSCRIPT debugger?

With the sync command the current source code will be loaded into the gateway and the existing will be overwritten. With the reload command just the already existing code in the gateway will be reloaded and reset and the debugger will jump into the first source code line. If there are differences recognized between the source code in the editor and the source code in the gateway, the debugger creates a "remote copy" version. Only this version can be debugged, but not edited.

What does the hint „remote copy" mean in the debugger's editor?

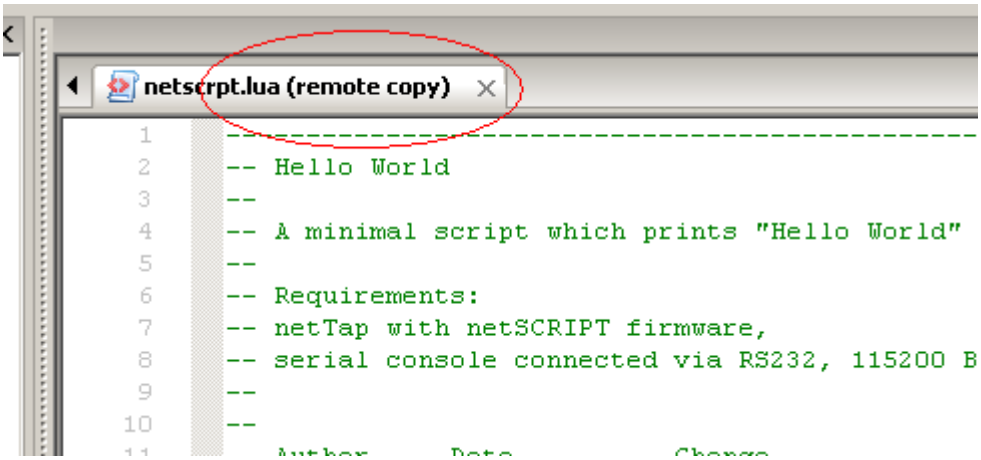


Figure 3: Indication ‚remote copy' in the netSCRIPT Debugger

The remark indicates that the shown source code is the one that is currently loaded in the gateway. The debugger creates this copy in case there are differences recognized between the source code in the gateway and the current source in the editor. The "remote copy" can not be edited but used for debugging purposes only. In order to harmonize use the sync command to load in the new source code.

Will the cyclic IO communication to the superordinated IO network be stopped in case the script is stopped?

No. Both processes are running fully independent of each other. netSCRIPT and its program sequence is never capable to stop or to interrupt the cyclic communication. The only possible accesses from netSCRIPT are encapsulated in the functions `BusIOxx()` via internal buffers.

2.5 Question to Optimizations

Is there a possibility to program a state machine via functions?

Like variables also functions are handled internally as tables. A function thus has a name and a value. This value can be changed during runtime and hence decide which function will be called next.

Example code:

```
if myfunc then
  myfunc()
else
-- defining all subfunctions
function my_state_init_done()
  ..
  myfunc = my_state_1
end

function my_state_1()
  ..
  myfunc = my_state_2
end

function my_state_2()
  ..
  myfunc = my_state_init_done
end

myfunc = my_state_init_done
end
```

Is a chained `if-elseif` conditional check proceeding all checks?

No, the netSCRIPT interpreter optimizes. A sequence of `if-elseif` inquiries will be left with the first met condition. The rest will be bypassed. The script executing will be slightly faster, if conditions are being placed in the sequence that is more likely.

How does a local distinguish from a global variable?

Variables with preceding `local` are valid only until the executed block where they are defined. The use of local variable has a small speed advantage, since the interpreter accesses to local variables via indexes and not via their names in the variable table.

What is happening to local variables at the end of the script?

Local variables defined in the root of script are no longer available in the next cycle of the script.

2.6 Question to String Operations

How can variables of the type `string` be merged together?

With the operator `..`

Example code:

```
string_a = “Hello“
string_b = “World“
result = string_a .. string b .. “2010“
-- will result in string “Hello World 2010“
```

How do I get the length of a string?

With the `:len()` operator and preceding string variable

Example code:

```
string_a = “Hello“
len = string_a:len()
-- will result in value 5 for variable len
```

A part of a string variable shall be assigned to another variable, how does it work?

Use the `string.sub()` operator. You need to specify the first and the last character that shall be cut out. It starts counting with 1. Negative values counting characters in the string from behind, starting with -1 as last character.

Example code:

```
string_a = “Hello“
extract_a = string.sub(string_a,2,4)
-- will result in value “ell“ for variable extract
extract_b = string.sub(string_a,2,-2)
-- same result
```

Usally the serial control commands are outside the alphabet, how do I declare those in a serial stream?

Use the `string.char()` operator and with one or more decimal or hexadecimal values.

Example code:

```
STX = string.char(0x02)
CR_LF = string.char(13, 10)
```

How to convert a string to a numeric value?

Use the `string.byte()` operator.

Example Code:

```
string.byte(”a”) -- will result a numeric value 97
string.byte(”abcde”,2) –- will result a numeric value 98 (”b”)
string.byte(”abcde”,-1) –- will result a numeric value 101 (”e”)
```

2.7 Questions to the Serial Communication

Is it necessary to hand over all configuration parameter when the port opening function is called?

No. The function operates with default values. Only those will be changed that a passed over. A simple `PortOpen()` with no parameters configures the serial port to 115200 Baud, 8, N, 1.

You can enter serial parameter in the SYCON.net configuration tool, how do I access them in the script?

Use the function `PortReadConfigDb()` and assign the return value to a variable.

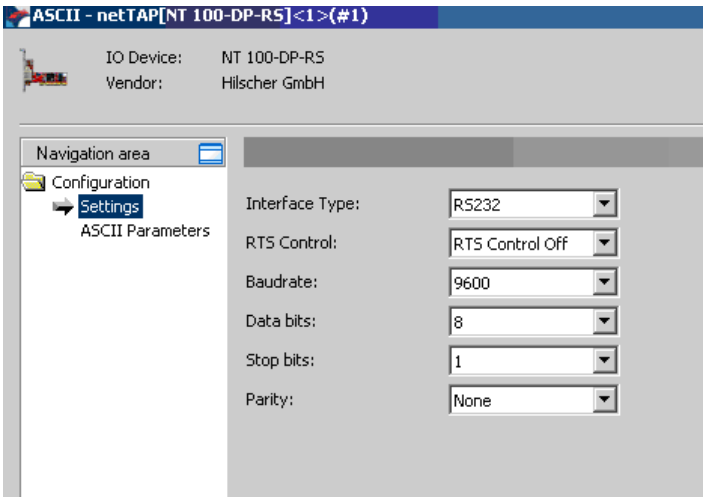


Figure 4: Serial Communication Parameters

Example code:

```
conf = PortReadConfigDb()
```

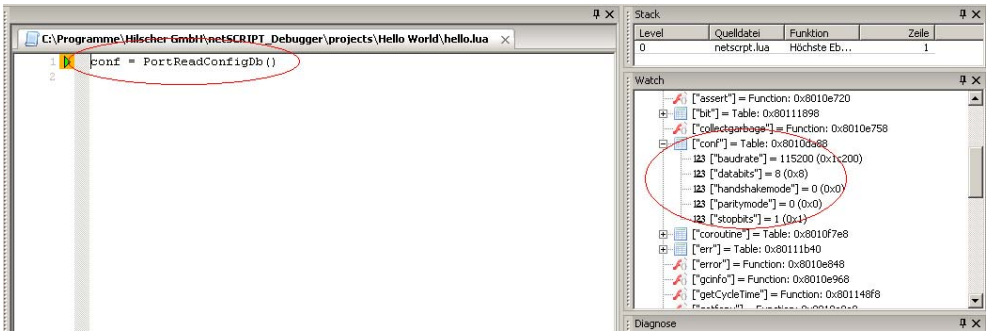


Figure 5: Example PortReadConfigDb()

Are those serial communication parameters that have been configured in SYCON.net changeable in the script?

Yes. Just take the values and assign them to a variable and modify those of interest.

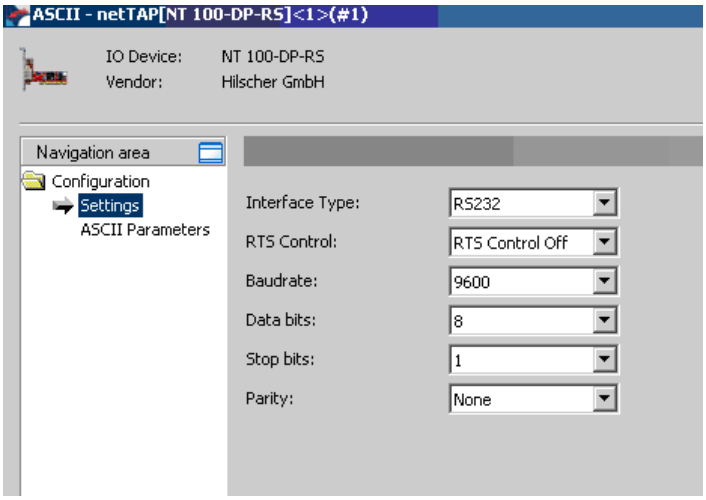


Figure 6: Serial Communication Parameters

Example code:

```
conf = PortReadConfigDb()
conf.baudrate = 9600 -- set baudrate to 9600baud in any case
myport = PortOpen(conf)
```

After sending a telegram the UART shall switch to receive mode immediately, which function needs to be used?

In cases where the time between the two calls of the functions `:PortSend()` and `:PortReceive()` is not sufficient to guarantee immediate receive-ready function especially at high baud rates, the combining function `:PortExchange()`has to be used.

2.8 Questions about Communication to the superordinated IO Network

Where shall the port opening function `BusIOOpen()` be placed in the script to initialize communication from netSCRIPT to the IO network?

In the initialization phase – in a part of your program that is executed only once. Dividing a program into parts can be realized using a state variable and a conditional check.

Is it possible to open another IO port?

No, it is only allowed to open one port at a time.

Is it possible to close the IO port and reopen it again?

Yes, as often as necessary.

Should the initialization ready event be reported to the IO controller?

The use of the function `:BusIOSetRun()` is optional. The report has the following advantage: After a power on all input data to the controller are all zero. Setting the run-bit indicates the controller that netSCRIPT has initialized well and is ready for communication.

Is it necessary to release transmit and receive function before activating a command from the IO controller?

No, not immediately. A write command can be activated from the controller without a release, but in netSCRIPT both functions `:BusIORead()` or `:BusIOWrite()` do not work until the they have been released by the IO controller. Just setting the release bits `APP_HS_TX_ENABLE_CMD` and `APP_HS_RX_ENABLE_CMD` in the IO synchronization register enables the function. If the bits are cleared both functions are freezed.

Are the release bits `APP_HS_TX/RX_ENABLE_CMD` in the synchronization register automatically acknowledged by netSCRIPT in the corresponding synchronization register?

Yes. Setting the bits `APP_HS_TX_ENABLE_CMD` or `APP_HS_RX_ENABLE_CMD` will be automatically acknowledged by netSCRIPT by setting the bits `PROT_HS_TX_ENABLE_ACK` or `PROT_HS_RX_ENABLE_ACK`. A precondition is that the IO port has been opened before using the function `BusIOOpen()`.

Is the usage of the error reporting function to the superordinated network optional?

Yes. The function `:BusIOSetError()` is optional

Are the error codes custom?

Yes, there are no restrictions. The full range from 0 to 2^31-1 can be used.

In case the IO controller sends a reset request, is it really necessary to perform a reset then?

No. Of course the functionality has been created to realize a reset command, but it can be used for any other purpose too. There are not restrictions.

When do I use the read function `:BusIORead()` without acknowledgement?

Reading data from the IO network without acknowledgement is used in cases where processing the data within netSCRIPT needs more time. You avoid then getting a new command before the old data has been processed. At the end of processing an additional call of `:BusIORead()` with acknowledgment option finally confirms the data as being read.

How do you hand over errors synchronous to a read confirmation?

There does not exist a handshake between the IO controller and netSCRIPT concerning errors. The error bits in the synchronization register to the controller can be set at any time within netSCRIPT. But if the acknowledgement a read command shall be synchronous to an error report you can use following trick. Report the error before you acknowledge the reception of new read data. This enables the IO controller to check for errors always when it sees the data as being confirmed.

Example code:

```
mystring = myport:BusIORead(false) -- read data, no confirm
… -- work with the data
myport:BusIOSetError( “receive”,true, myerrorcode)
myport:BusIORead(true) -- confirm data after setting error
```

2.9 Questions to the IO Controller

Via a SIMATIC-S7 program the transmit and receive release bits shall be set to enable send and receive communication, how does it work?

The bits 7 and 6 in the output synchronization register of the PLC have to be set. The example assumes that the output synchronization register is located at MB 0.

Example code:

```
L     B#16#C0       // load hex 0xC0, bit 7 and 6 being set
T     MB 0          // set bitmask to address 0
```

The number transmitted IO data exceeds the length of 4 bytes using a SIMATIC-S7, what shall I do?

The special functions SFC14 and SFC15 for handing over the data have to be used to guarantee consistent data transmission.

Example code SFC15:

```
CALL  "DPWR_DAT"   // call output write function SFC15
 LADDR  :=W#16#0   // start at output address 0
 RECORD :=P#M 0.0 BYTE 20 // copy from merker 0
 RET_VAL:=MW200           // where to return function's error
```

Example code SFC14:

```
CALL  "DPRD_DAT"  // call input read function SFC14
 LADDR  :=W#16#0  // start at input address 0
 RET_VAL:=MW200   // where to return function's error
 RECORD :=P#M 20.0 BYTE 20 // copy to meker 20
```

How do I realized the recognition of a write command coming from netSCRIPT in a SIMATIC-S7 program?

The example assumes that the input and output synchronization register is located at MB 0 and MB 20

Example code:

```
  L MB 0    // load output handshake byte
  L MB 20   // load input handshake byte
  XOW       // do XOR knotting
  L 2       // load bit mask for APP_HS_RX_CMD = new command?
  UW        // make AND knotting and test for bit
  SPZ label // jump if result is zero, no command
```

How do I acknowledge a write command from netSCRIPT in a SIMATIC-S7 program?

The example assumes that the output synchronization register is located at MB 0.

Example code:

```
   L MB 0   // load current handshake byte
   L 2      // load bit mask APP_HS_RX_ACK, acknowledge
   XOW      // flip bit
   T MB 0   // transfer handshake byte for next cycle
```

2.10 Questions to configurable Variables from outside

In the configuration tool SYCON.net exists the possibility to define variable outside the script, how can I access to those variables in my script?

The variables are held in the global table VAR and the configured values are assigned to them. Via the access path `VAR.Tabellenname.Variablenname` you can access to these values at any time in the script.

The screen shot shows the variables `mynumber`, `mybool` and `mystring` along with their values in the table `mytable`.

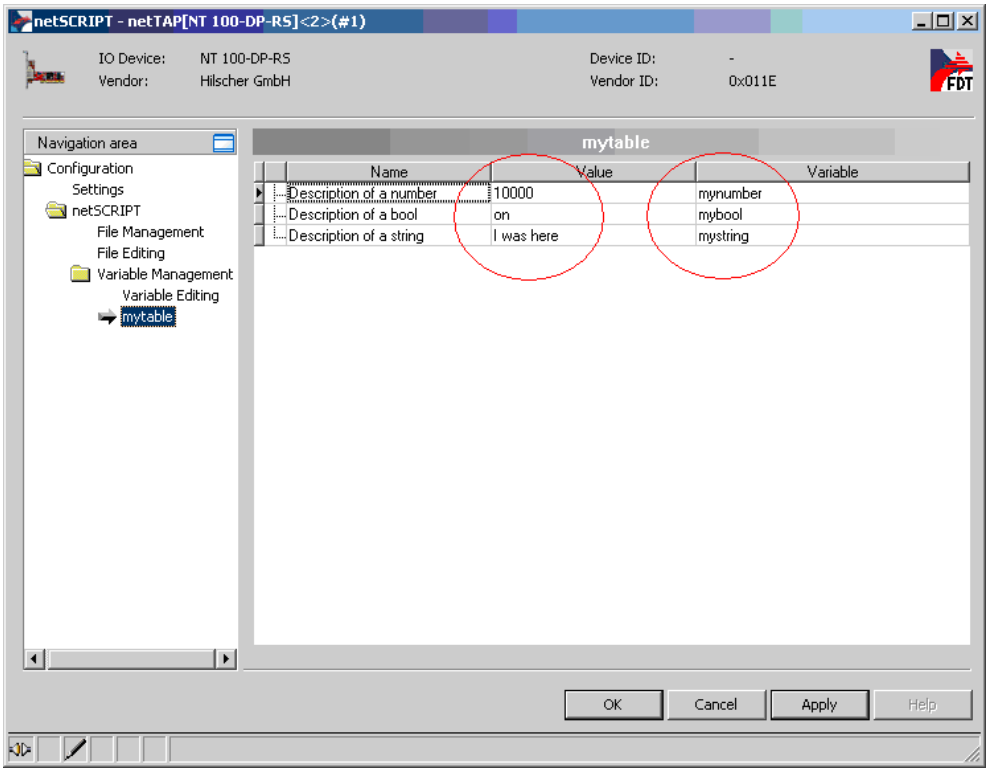


Figure 7: Configurable Variable from outside

Example code:

```
strlocal = VAR.mytable.mystring      -- equals to “I was here“
```

If the variables are defined from outside in SYCON.net and the netSCRIPT debugger is used afterwards, do these variables still exist?

Yes. The variables are stored in a special configuration file in the gateway. To this file netSCRIPT debugger and the script file have read access only.

3 Appendix

3.1 List of Figures

3.2 List of Tables

3.3 Contacts

Headquarters

Germany
Hilscher Gesellschaft für
Systemautomation mbH
Rheinstrasse 15
65795 Hattersheim
Phone: +49 (0) 6190 9907-0
Fax: +49 (0) 6190 9907-50
E-Mail: info@hilscher.com

Support
Phone: +49 (0) 6190 9907-99
E-Mail: de.support@hilscher.com

Subsidiaries

China
Hilscher Ges.f.Systemaut. mbH
Shanghai Representative Office
200010 Shanghai
Phone: +86 (0) 21-6355-5161
E-Mail: info@hilscher.cn

Support
Phone: +86 (0) 21-6355-5161
E-Mail: cn.support@hilscher.com

France
Hilscher France S.a.r.l.
69500 Bron
Phone: +33 (0) 4 72 37 98 40
E-Mail: info@hilscher.fr

Support
Phone: +33 (0) 4 72 37 98 40
E-Mail: fr.support@hilscher.com

India
Hilscher India Pvt. Ltd.
New Delhi - 110 025
Phone: +91 11 40515640
E-Mail: info@hilscher.in

Italy
Hilscher Italia srl
20090 Vimodrone (MI)
Phone: +39 02 25007068
E-Mail: info@hilscher.it

Support
Phone: +39/02 25007068
E-Mail: it.support@hilscher.com

Japan
Hilscher Japan KK
Tokyo, 160-0022
Phone: +81 (0) 3-5362-0521
E-Mail: info@hilscher.jp

Support
Phone: +81 (0) 3-5362-0521
E-Mail: jp.support@hilscher.com

Korea
Hilscher Korea Inc.
Suwon-Si, 443-810
Phone: +82-31-204-6190
E-Mail: info@hilscher.kr

Switzerland
Hilscher Swiss GmbH
4500 Solothurn
Phone: +41 (0) 32 623 6633
E-Mail: info@hilscher.ch

Support
Phone: +49 (0) 6190 9907-99
E-Mail: ch.support@hilscher.com

USA
Hilscher North America, Inc.
Lisle, IL 60532
Phone: +1 630-505-5301
E-Mail: info@hilscher.us

Support
Phone: +1 630-505-5301
E-Mail: us.support@hilscher.com