

Program your own serial protocol

- With open language for increased added value

Information about the author



Name:	Armin Beck
Company:	Hilscher GmbH
Item:	Product manager
Address:	Rheinstraße 15
Postcode, City:	65795 Hattersheim
Fon:	06190-9907-849
Fax:	06190-9907-50
Email:	abeck@hilscher.com

Nowadays, gateways are used to couple devices with proprietary serial protocols to a standardised industrial bus. At Hilscher customers implement their protocol themselves using a scripting language without having to fall back on cost-intensive programmers and services. To accomplish this, the "netSCRIPT" solution consciously takes advantage of the Lua open-source script language as the basis. Thanks to the extremely simple language syntax, a small footprint and its speed, Lua has established itself as the most widely propagated scripting language on the market in the last 20 years. Once learned, the open language provides greater added value throughout the entire company, whereas proprietary gateway programming solutions with unknown programming language preclude further use.



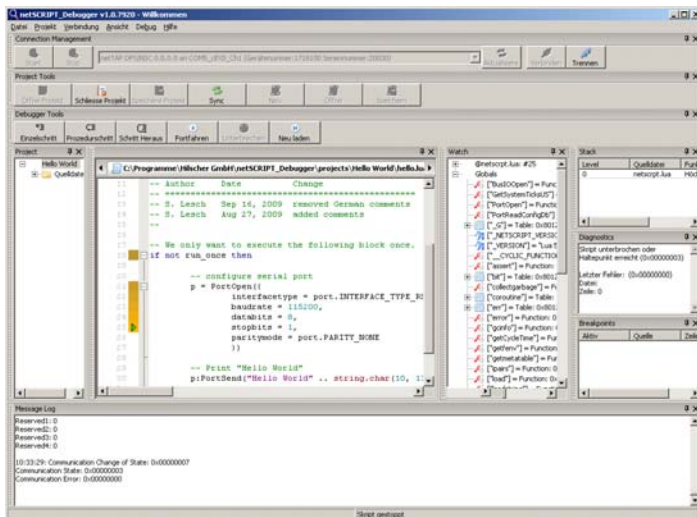
Programmable gateway

With its "ease", the Lua language is even suited to those who do not have programming experience. The basic knowledge from schooldays is sufficient to master it. The language is today predominantly used to utilise applications or libraries that were professionally written in high-level languages such as C/C++/C# via a Lua frontend. While doing so, the embedded script integrates with the core software, enabling simple and fast use of its functions. Whether it's Win32, Linux, Unix, MAC OS X or android; there are Lua interpreters available with complete graphics libraries, file system access functions, Ethernet Socket Interface or web server for nearly all platforms for implementing perfectly suited, typical applications in company processes. For instance, automated tests can be effectively written to optimize production processes, which collect the accruing measurements and characteristic

values for quality assurance and traceability; another example is the ability to check the behaviour of new software components per script template for their compatibility with their predecessors.

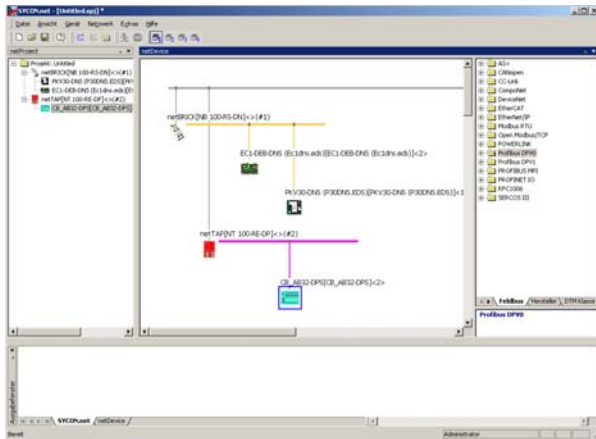
netSCRIPT expands Lua with serial interface communication functions and I/O data access functions to the higher-level network. The standard mathematical Lua library facilitates calculating complex algorithms for control paths right in the device. With a 2Mbyte script memory, entire communications profiles can be mapped and complete applications can be realized. What up to now could only be implemented with great expenditure in the PLC is now taken care of by the intelligent gateway.

Data access in the script is solely granted through variables; indirect addressing through pointers such as C no longer exists. That protects against unauthorised and unwanted memory accessing, securing the entire gateway application against unpredictable system crashes. The script is programmed with the netSCRIPT development environment. The debugger checks for correct program sequencing. All program sections and variables can be systematically examined step-wise or procedural.



netSCRIPT Development environment

The final and tested script is loaded into the gateway together with the higher-level bus configuration through the common FDT/DTM based network configuration tool SYCON.net. At each power cycle the gateway immediately starts with the periodical script processing. The cycle time can be scaled in milliseconds. A special feature permits creating variables and pre-assigning them in the tool with values which are made available to the script as online parameters after loading. For instance, key parameters such as a baud rate can be easily set and changed by an end user without any need for him to be familiar with the script and its sequence.



SYCON.net bus configuration tool

Experience has shown that even a demanding serial protocol can be implemented with netSCRIPT in one day. An example of a typical program skeleton demonstrates how easy it is to use netSCRIPT:

```

-- Open UART and IO bus interfaces
uart = PortOpen()
io    = BusIOOpen()
...
-- Load output data and transmit
output = io:BusIORead()
if output then
    uart:PortSend(output)
end
...

```

Script extract - reading output data and serial transmission

...Open() opens the access to the serial and higher-level I/O bus. The read command BusIORead() reads the I/O bus output data and, via PortSend(), the data is transmitted through the serial interface. In between the "output" data could be modified at will in the script.

Providing an open-source based programming solution is in line with the general trend in automation. Device manufacturers are starting to develop products based on Linux - indeed even publishing their own source codes. The collective community then works on product improvements, which saves development costs. Conversely, the user receives a flawless, user-friendly product that has been tried and proven. In this sense thanks to Lua, netSCRIPT is a powerfully and flexibly designed programming solution right from the start. If employees in the company are already familiar with Lua, the learning curve is reduced. Other companies, for which Lua is new, profit from the versatile offer of freely accessible Lua applications with a high recycling factor in additional business processes.

Please visit www.hilscher.com, Gateways/netTAP product section, for more information.