SDK

NetModule routers are shipping with a Software Development Kit (SDK) which offers a simple and fast way to implement customer-specific functions and applications. It consists of:

1. An SDK host which defines the runtime environment (a so-called sandbox), that is, controlling access to system resources (such as memory, storage and CPU) and, by doing so, catering for the right scalability.
2. An interpreter language called arena, a light-weight scripting language optimized for embedded systems, which uses a syntax similar to ANSI-C but adds support for exceptions, automatic memory management and runtime polymorphism on top of that.
3. A NetModule-specific Application Programming Interface (API), which ships with a comprehensive set of functions for accessing hardware interfaces (e.g. digital IO ports, GPS, external storage media, serial ports) but also for retrieving system status parameters, sending E-Mail or SMS messages or simply just to configure the router.

Anyone, reasonably experienced in the C language, will find an environment that is easy to dig in. However, feel free to contact us via router@support.netmodule.com and we will happily support you in finding a programming solution to your specific problem.

The Language

The arena scripting language offers a broad range of POSIX functions (like printf or open) and provides, together with tailor-made API functions, a simple platform for implementing any sort of applications to interconnect your favourite device or service with the router. Here comes a short example:

example.are

```
/* We are going to eavesdrop on the first serial port
* and turn on lights via a digital I/O output port ,
* otherwise we 'd have to send a short message .
*/
for ( attempts=0; attempts<3; attempts ++) {
  if ( nb_serial_read ("serial0") == "Knock Knock !") {
    nb_serial_write ("serial0", "Who 's there ?");
    if ( nb_serial_read ("serial0") == " Santa ") {
      printf (" Hurray !\n");
      nb_dio_set ("out1", 1);
    }
  }
}
nb_sms_send ("+123456789", "No presents this year :(")
```

A set of example scripts can be downloaded directly from the router, you can find a list of them in the

NetModule AG
Meriedweg 11
3172 Niederwangen
T +41 31 985 25 10
F +41 31 985 25 11
Switzerland

NetModule GmbH
Frankfurter Strasse 92
65760 Eschborn
T +49 6196 77 99 79 0
F +49 6196 77 99 79 9
Germany

appendix. The SDK language manual gives a detailed introduction of the language, including a description of all available functions.

SDK API Functions

The current range of API functions can be used to implement the following features:

1. Send/Retrieve SMS
2. Send E-mail
3. Read/Write from/to serial device
4. Control digital input/output ports
5. Run TCP/UDP servers
6. Run IP/TCP/UDP clients
7. Access files of mounted media (e.g. an USB stick)
8. Retrieve status information from the system
9. Get or set configuration parameters
10. Perform config/software updates
11. Write to syslog
12. Transfer files over HTTP/FTP
13. Get system events / Reboot system
14. Control the LEDs
15. Get system events or reboot system
16. Scan available networks
17. Web Pages
18. Voice control functions
19. SNMP functions
20. CAN socket functions
21. MODBUS functons
22. Various network-related functions
23. OPC-UA functions (coming soon)
24. Encode functions (coming soon)
25. Other system-related functions

The SDK API manual provides an overview but also explains all functions in detail.

Please note that some functions require the corresponding services (e.g. E-Mail, SMS) to be properly configured prior to utilizing them in the SDK. Let us now pay some attention to the very powerful API function `nb_status`. It can be used to query the router's status values in the same manner as they can be shown with the CLI. It returns a structure of variables for a specific section (a list of available sections can be obtained by running `cli status -h`).

By using the dump function you can figure out the content of the returned structure:

sample2.are

```
/* dump current location */
dump ( nb_status (" location "));
```

The script will then generate lines like maybe these:

NetModule AG
Meriedweg 11 T +41 31 985 25 10 Switzerland
3172 Niederwangen F +41 31 985 25 11

NetModule GmbH
Frankfurter Strasse 92 T +49 6196 77 99 79 0 Germany
65760 Eschborn F +49 6196 77 99 79 9

sample3.are

```
struct(8): {
.LOCATION_STREET = string[11]: "Bahnhofquai"
.LOCATION_CITY = string[10]: "Zurich"
.LOCATION_COUNTRY_CODE = string[2]: "ch"
.LOCATION_COUNTRY = string[11]: "Switzerland"
.LOCATION_POSTCODE = string[4]: "8001"
.LOCATION_STATE = string[6]: "Zurich"
.LOCATION_LATITUDE = string[9]: "47.3778058"
.LOCATION_LONGITUDE = string[8]: "8.5412757"
}
```

In combination with the `nb_config_set` function, it is possible to start a re-configuration of any parts of the system upon status changes. Here is an example how one might adopt those functions:

sample4.are

```
/* check current city and enable the second WAN link */
location = nb_status (" location ");
if ( location ) {
  city = struct_get ( location , " LOCATION_CITY ");
  if ( city == " Wonderland ") {
    for (led = 0; led < 5; led ++) {
      nb_led_set (led , LED_BLINK_FAST | LED_COLOR_RED );
    }
  } else {
    printf ("You 'll never walk alone in %s ...\ n", city );
    nb_config_set ("wanlink.1.mode=1");
  }
}
```

Running SDK

In the SDK, we are speaking of scripts and triggers which form jobs. Any arena script can be uploaded to the router or imported by using dedicated user configuration packages. You may also edit the script directly at the Web Manager or select one of our examples. You will further have a testing section on the router which can be used to check your syntax or doing test runs. Once uploaded, you will have to specify a **trigger**, that is, telling the router when the script is to be executed. This can be either time-based (e.g. each Monday) or triggered by one of the pre-defined system events (e.g. wan-up). With both, a script and a trigger, you can finally set up an SDK **job** now. The test event usually serves as a good facility to check whether your job is doing well. The admin section also offers facilities to troubleshoot any issues and control running jobs. The SDK host (sdkhost) corresponds to the daemon managing the scripts and their operations and thus avoiding any harm to the system. In terms of resources, it will limit CPU and memory for running scripts and also provide a pre-defined

NetModule AG
Meriedweg 11
3172 Niederwangen
T +41 31 985 25 10
F +41 31 985 25 11
Switzerland

NetModule GmbH
Frankfurter Strasse 92
65760 Eschborn
T +49 6196 77 99 79 0
F +49 6196 77 99 79 9
Germany

portion of the available flash storage. You may, however, extend it by external USB storage or (depending on your model) SD cards. Files written to /tmp will be hold in memory and will be cleared upon a restart of the script. As your scripts operate in the sandbox, you will have no access to tools on the system (such as `ifconfig`).

Simple How-To & Testing

- Testing the SMS send script

Useful SDK Examples

Web Manager

- Custom Configuration Parameters
- Web GUI extension
- Webpage Extention to send a Pulse on the Digital Output

Configurations

- Set fake default route dynamically based on SSID
- Start softwareupdate if newer version is available
- Request a unique remote config-file from a fileserver and delete it afterwards

Status Parameters

- Connection Statistics
- Get status values via SNMP
- Monitor and reset data usage on mobile network
- Sends status values to a remote UDP server

Supervision

- Supervising a VPN Tunnel
- Monitor the system load and put a warning in the log file if critical
- Create a Phonecall Alarm
- Log Status Values to Syslog
- Change WWAN Hotlink according to the Service Type

GPS

- Broadcast selected GPS NMEA data to different UDP ports
- Send GNSS data to a remote UDP port in $GP instead of $GN format

Serial

- Serial Point to Multipoint
- Write the GPS NMEA Frames to the Serial Interface

NetModule AG
Meriedweg 11 T +41 31 985 25 10 Switzerland
3172 Niederwangen F +41 31 985 25 11

NetModule GmbH
Frankfurter Strasse 92 T +49 6196 77 99 79 0 Germany
65760 Eschborn F +49 6196 77 99 79 9

- Read data from the serial port and write it to files in the flash
- Write non ASCII String to the Serial Port

HTTP

- Forward File Content of Files on FTP Server via SMS
- Webservices using HTTP GET
- Download a file from a HTTP Server and store it to a USB flash drive

Modbus

- Access to the digital I/Os via Modbus TCP
- Read out a Modbus TCP Temperatur Sensor and send an alarm Email
- Read out a Janitza UMG511 via Modbus TCP

MQTT

- Send Router Status Informations to an MQTT Broker
- Send Router Status Informations to an MQTT Broker - Advanced Version

Wifi / WLAN

- Change Wifi Mode based on SSID

Various

- Run a command using Telnet
- Sending SMS via TCP
- Publish OPC-UA datas to the Web as JSON-Object also monitoring values

Helpful Functions

- Convert IBIS Telegrams to valid ASCII Strings (Umlaute)
- Cast two Modbus Register on to one float variable

Built-in Scripts

Script	Description
best-operator.txt	This script will scan for operators on startup and choose the one with the best signal
candump.txt	This script can be used to receive CAN messages
config-summary.txt	This script shows a summary of the currently running configuration.

NetModule AG
Meriedweg 11 T +41 31 985 25 10 Switzerland
3172 Niederwangen F +41 31 985 25 11

NetModule GmbH
Frankfurter Strasse 92 T +49 6196 77 99 79 0 Germany
65760 Eschborn F +49 6196 77 99 79 9

Script	Description
dio-monitor.txt	This script monitors the DIO ports and sends a SMS to the specified phone number.
dio-server.txt	This script implements a TCP server which can be used to control the DIO ports.
dio.txt	This script can be used to set a digital output port.
dynamic-operator.txt	This script will scan Mobile2 and dial the appropriate SIM on Mobile1
email-to-sms.txt	This script implements a lightweight SMTP server which is able to receive mail and forward them as SMS to a phone number.
etherwake.txt	This script can be used to wake up a sleeping host (WakeOnLan)
gps-monitor.txt	A script for activating WLAN as soon as GPS position (lat,lon) is within a specified range.
gps-udp-client-compat.txt	This script sends the local GPS NMEA stream (incl. serial/checksum) to a remote UDP server.
gps-udp-client.txt	This script sends the local GPS NMEA stream to a remote UDP server.
gps-upd-client-compat.txt	This script sends the local GPS NMEA stream to a remote UDP server (incl. device identity).
led.txt	This script can be used to set a LED
modbus-rtu-master.txt	This script can be used to read messages from the serial port.
modbus-rtu-slave.txt	This script implements a modbus slave server
modbus-tcp-rtu-gateway.txt	This script can be used to read messages from the serial port.
mount-media.txt	This script can be used to mount an USB storage stick.
opcua-browse.txt	This script can be used to browse the Addressspace of an OPC-UA-Server. (coming soon)
opcua-read.txt	This script can be used to read the value from a Node at a OPC-UA-Server. (coming soon)
opcua-search.txt	This script can be used to search for some Nodes at a OPC-UA-Server. (coming soon)
opcua-write.txt	This script can be used to write a new value to a Node at a OPC-UA-Server. (coming soon)
ping-supervision.txt	This script will supervise a specified host.
read-config.txt	This script can be used to read a configuration parameter.

NetModule AG
Meriedweg 11
3172 Niederwangen
T +41 31 985 25 10
F +41 31 985 25 11
Switzerland

NetModule GmbH
Frankfurter Strasse 92
65760 Eschborn
T +49 6196 77 99 79 0
F +49 6196 77 99 79 9
Germany

Script	Description
remote-mail.txt	This script reads and sends mails from a remote IMAP/POP3/SMTP server
scan-mobile.txt	This script can be used to switch the Mobile LAI according to available networks
scan-wlan.txt	This script can be used to switch the WLAN client network according to availability
send-mail.txt	This script will send an E-Mail to the specified address.
send-sms.txt	This script will send an SMS to the specified phone number.
serial-read.txt	This script can be used to read messages from the serial port.
serial-readwrite.txt	This script will write to and read from the serial port.
serial-tcp-broadcast.txt	This script reads messages coming from the serial port and forwards them via TCP to remote hosts (and vice versa).
serial-tcsetattr.txt	This script can be used to set/get the attributes of the serial port.
serial-udp-server.txt	This script reads messages coming from the serial port and forwards them via UDP to a remote host (and vice versa).
serial-write.txt	This script can be used to write a message to the serial port.
set-ipsec-route.txt	set route to IPSEC server depending on active WWAN / WLAN network
sms-control.txt	This script will execute commands received by SMS.
sms-delete-inbox.txt	This script can be used to flush the SMS inbox.
sms-read-inbox.txt	This script can be used to read the SMS inbox.
sms-to-email.txt	This script will forward incoming SMS messages to a given E-mail address.
sms-to-serial.txt	This script can be used to write a received SMS to the serial port.
snmp-agent.txt	This script extends MIB entries of the SNMP agent
snmp-cmd.txt	This script issues SNMP set/get commands
snmp-trap.txt	This script can be used to send SNMP traps
status.txt	This script can be used to display all status variables
syslog.txt	Throw a simple syslog message.
tcpclient.txt	This script sends a message to a TCP server.

NetModule AG
Meriedweg 11
3172 Niederwangen
T +41 31 985 25 10
F +41 31 985 25 11
Switzerland

NetModule GmbH
Frankfurter Strasse 92
65760 Eschborn
T +49 6196 77 99 79 0
F +49 6196 77 99 79 9
Germany

Script	Description
tcpserver.txt	This script implements a TCP server which is able to receive messages.
techsupport.txt	This transfers a techsupport to a remote FTP server
transfer-file.txt	This scripts archives a remote file
transfer.txt	This scripts stores the latest GNSS positions in a remote FTP file
udp-msg-server.txt	This script will run an UDP server which is able to receive messages and forward them as SMS/E-Mail.
udpclient.txt	This script sends a message to a remote UDP server.
udpserver.txt	This script implements an UDP server which is able to receive messages.
update-config.txt	This script can be used to perform a configuration update
voice-dispatcher-audio.txt	This script implements an audio voice dispatcher
webpage.txt	This script will generate a page which can be viewed in the Web Manager
write-config.txt	This script can be used to set a configuration parameter.

Access the Filesystem

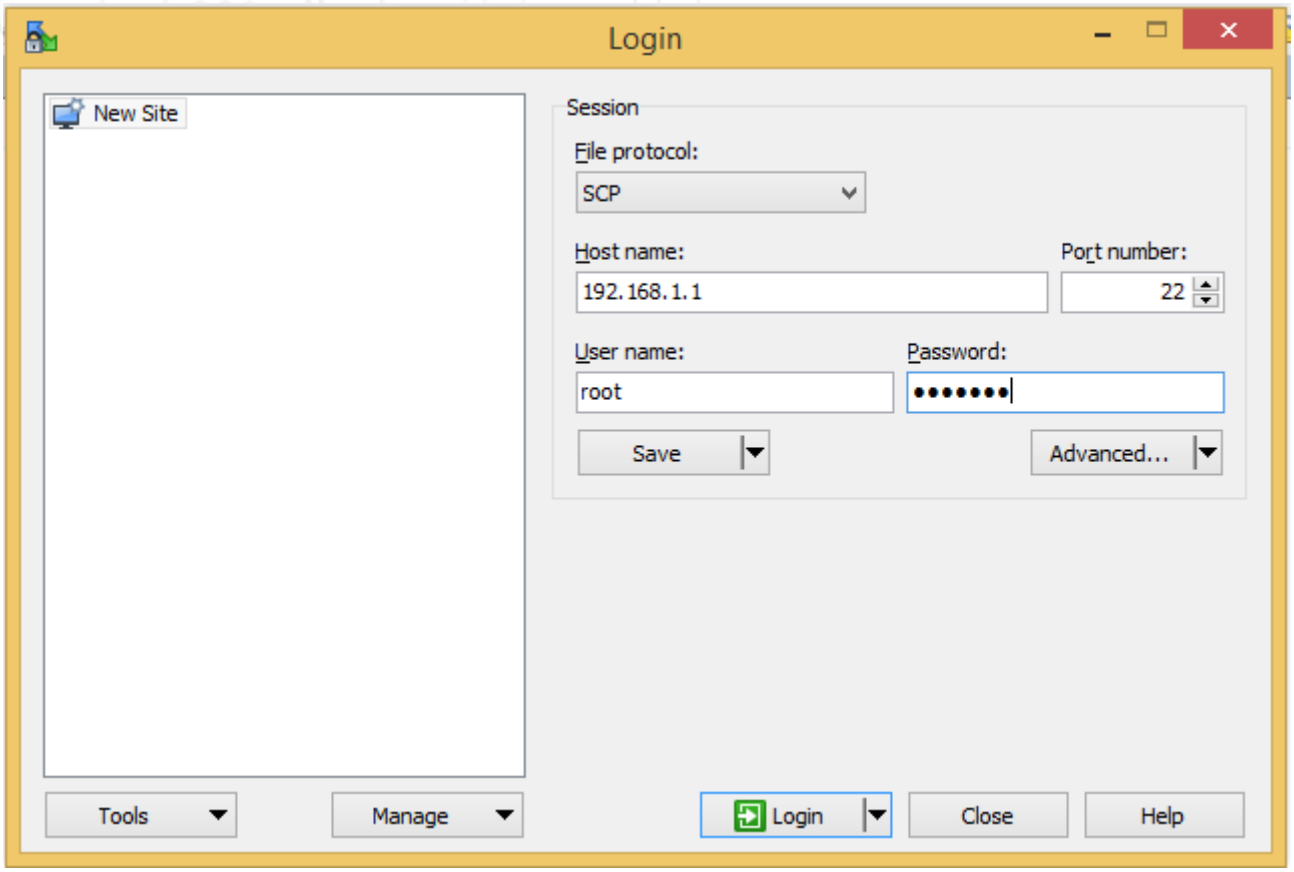
You can acces the SDK Filesystem externaly via SCP:

With Windows you can use the opensource software WinSCP

Connect with the Protocoal SPC, User “root” and your admin password:

NetModule AG
Meriedweg 11
3172 Niederwangen
T +41 31 985 25 10
F +41 31 985 25 11
Switzerland

NetModule GmbH
Frankfurter Strasse 92
65760 Eschborn
T +49 6196 77 99 79 0
F +49 6196 77 99 79 9
Germany



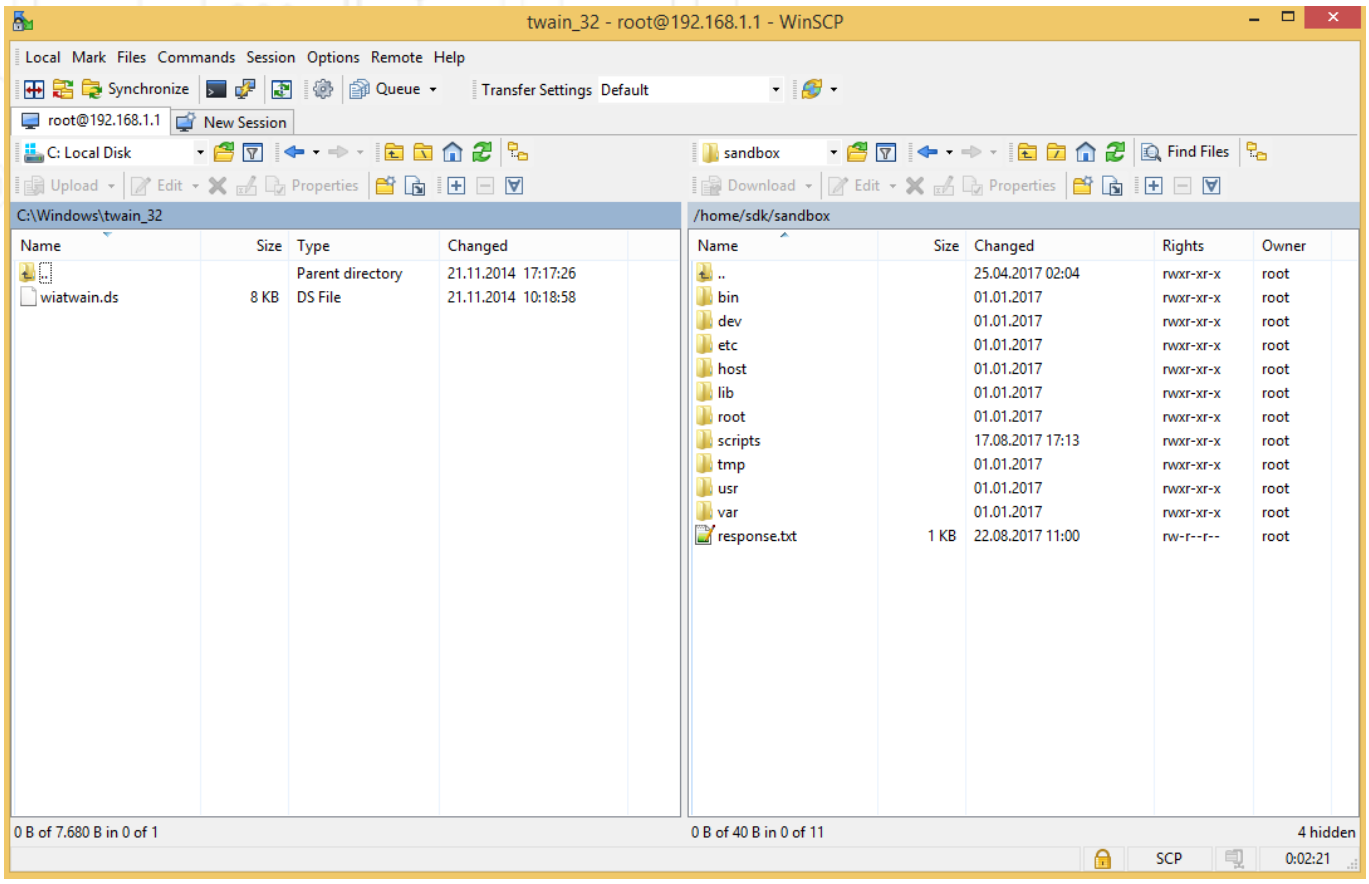
Please then go to the folder

```
/home/sdk/sandbox/
```

Where you find the Filesystem which is usable from the SDK

NetModule AG
Meriedweg 11
3172 Niederwangen
T +41 31 985 25 10
F +41 31 985 25 11
Switzerland

NetModule GmbH
Frankfurter Strasse 92
65760 Eschborn
T +49 6196 77 99 79 0
F +49 6196 77 99 79 9
Germany



NetModule AG
Meriedweg 11
3172 Niederwangen
T +41 31 985 25 10
F +41 31 985 25 11
Switzerland

NetModule GmbH
Frankfurter Strasse 92
65760 Eschborn
T +49 6196 77 99 79 0
F +49 6196 77 99 79 9
Germany