

Software Manual – Programming and Integration

Software Driver for Real Time Target

SEA 9741/9744/9745

4G / GPS Communication Module for CompactRIO™



SEA Science & Engineering
Applications Datentechnik
GmbH

Doc. No.: HB/SEA 974x SoftwareManual_RT/2.0/Jan-2017

Subject to modifications.

S.E.A. Datentechnik GmbH takes no responsibility for damage arising out of or related to this document or the information contained in it.

Product and company names listed are trademarks or trade names of their respective companies.

© Science & Engineering Applications Datentechnik GmbH

Address:

S.E.A. Datentechnik GmbH
Muelheimer Strasse 7
53840 Troisdorf
Germany

Phone: +49 2241 12737-0

Fax: +49 2241 12737-14

For support please contact the support email address:

techsupport@sea-gmbh.com

Contents

1	Change Notes.....	5
2	Getting Started.....	6
2.1	General.....	6
2.2	End User License Agreement (EULA).....	6
2.3	Symbols, Notations and Nomenclature.....	6
3	Software Packages.....	8
4	Installation.....	9
4.1	Hardware Requirements.....	9
4.2	Software Requirements.....	9
5	Quick Start.....	10
6	Description of Functionality.....	12
6.1	CompactRIO Platform.....	12
6.2	SEA 974x Functionality.....	12
6.2.1	GSM.....	12
6.2.2	Remote Access	14
6.2.2.1	Dynamic DNS.....	14
6.2.2.2	Port Forwarding.....	14
6.2.2.3	VPN.....	15
6.2.3	GPS.....	15
6.2.4	Power up, Reset, Recover.....	16
7	Configuration.....	17
7.1	Management.....	17
7.2	Network.....	18
7.3	Dynamic DNS.....	18
7.4	Remote Accesses.....	19
7.5	Radio Status.....	19
8	Programming.....	20
8.1	Migration.....	20
8.2	Examples.....	20
8.3	Application Programming Interface API.....	20
8.3.1	Close Connection.....	20
8.3.2	Get Configuration.....	20
8.3.3	Get Version.....	21
8.3.4	Open Connection.....	21
8.3.5	GPS - Enable/Disable Messages.....	22
8.3.6	GPS - Enable/Disable SBAS.....	22
8.3.7	GPS - Read Messages.....	23
8.3.8	GPS - Reset GPS.....	23
8.3.9	GPS - Set Timepulse Interval.....	23
8.3.10	GSM – Init GSM.....	24
8.3.11	GSM – Query IMEI.....	24
8.3.12	GSM – Query Network.....	25
8.3.13	GSM – Query Registration.....	25
8.3.14	GSM – Query Service Status.....	26

8.3.15	GSM – Query Signal Quality.....	26
8.3.16	GSM – Reset GSM.....	28
8.3.17	SMS – Delete SMS.....	28
8.3.18	SMS – Init SMS.....	28
8.3.19	SMS – Query SMS Service Center.....	29
8.3.20	SMS – Receive SMS.....	29
8.3.21	SMS – Send SMS.....	30
8.3.22	Packet - Close Packet Connection.....	30
8.3.23	Packet – Get External IP Address.....	31
8.3.24	Packet – Open Packet Connection.....	31
8.3.25	Packet - Query Packet Connection Status.....	32
8.4	Error Codes.....	32
9	Deployment.....	34
10	Trouble Shooting.....	35
A	Figures.....	37
B	Tables.....	38

1 Change Notes

#	Description	Changes
1	1.0	Initial release for new LabVIEW driver architecture (Module Development Kit 2.0, MDK2).
2	2.0	<ol style="list-style-type: none">1. Main Update release with a new set of driver functions for FPGA (GPS).2. Oldest supported LabVIEW Version is now 2013.

2 Getting Started

2.1 General

Before starting to work with the SEA 9741/9744/9745 module please read this document and the complete hardware manual carefully. If there are any questions about operating the module or if any term is not understood, please contact the vendor before using the module. Please check the download area on the S.E.A. website: <http://www.sea-gmbh.com> for updates of the manuals.



Refer to the hardware manual for details on operation instructions, safety guidelines and specifications for the SEA 9741/9744/9745 module.



Refer to the appropriate National Instruments™ documentation for details on National Instruments hardware.

We believe that all information in this manual is accurate. The document has been carefully reviewed for technical accuracy. In the event of technical or typographical errors, we reserve the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult the vendor if errors are suspected.





2.2 End User License Agreement (EULA)

Before operating the SEA 9741/9744/9745 module and the provided software you have to agree to the terms and conditions (EULA). This agreement is part of the software installation procedure. If you do NOT agree you can send back the hardware and software package within a period of two weeks after delivery. In this case S.E.A. will refund the product price and shipping costs.

In addition, the terms and conditions are available through the LabVIEW menu after installation (Tools > SEA > SEA 974x > Legal Information).

2.3 Symbols, Notations and Nomenclature

To improve clarity specific structuring elements (symbols) are used which have the following meaning:

Symbol	Meaning
<i>Names</i>	Specific names are printed using an <i>italic font</i>
[Text]	Place holders are marked by squared brackets
	Locations (paths, menus, URLs...) are printed using a <code>courier font</code>
	The yellow sign highlights important notes and warnings
	The blue mark highlights tips
	Reference to other documents



The notation SEA 974x is a short cut for a family of modules. If you read SEA 974x you can substitute it with the name of your particular type (SEA 9741/9744/9745). If any information contained in this manual refers exclusively to a particular module type, then this particular module name is used instead of SEA 974x.

For a better understanding, a short list of used terms will be given:

Name	Meaning
ACC	A ccuracy
API	A pplication P rogramming I nterface
DOP	D ilution of P recision
FPGA	F ield P rogrammable G ate A rray
GPS	G lobal P ositioning S ystem
GSM	G lobal S ystem for M obile C ommunications (GSM) denotes the second generation of mobile networks (2G) but in scope of this manual it is used as a generic term for mobile communication related features including all supported mobile networks like GSM, GPRS, UMTS, HSPA or LTE.
IRIG-B	Time code format standard of the Inter Range Instrumentation Group
LND	L ocal N etwork D evice is the device that is locally connected to the module (via the Ethernet port when using SEA 974x or USB port when using SEA 975x). This is the CompactRIO.
MDK2	M odule D evelopment K it, version 2 . MDK2 is a framework by National Instruments for developing 3 rd party cRIO modules.
MNO	M obile N etwork O perator. This the the provider of the mobile network like T-Mobile, AT&T, NTTDocomo or many others.
NI	N ational I nstruments
NI MAX	NI M easurement and A utomation E xplorer
NMEA	Is a standard protocol used by GPS receivers to transmit data (more precisely NMEA-0183) from the N ational- M arine- E lectronics- A ssociation.
OS	O perating S ystem
RT	R eal T ime
SBAS	S atellite B ased A ugmentation S ystem
SEA 974x	Generic term for module types: SEA 9741/9744/9745
VI	V irtual I nstrument
VIPM	V i P acket M anager

Tab. 1: Nomenclature

3 Software Packages

For the SEA 974x module there are 3 software packages available:

1. Driver Software for Real Time target

This driver software installs all resources required to access the module directly from the Real Time target through the Ethernet (SEA 974x) or USB (SEA 975x) connection. This driver software provides access to all module's features.

2. Driver Software for FPGA target

This driver software installs resources required to access the module directly from the FPGA target through the backplane connection. This driver software provides access to the GPS telemetry data only (offering a sub set of the Real Time driver functionality).

3. Module's Firmware (SEA 974x) / Real Time Support (SEA 975x)

Module's Firmware (SEA 974x): This software runs on the module. The module is shipped with a firmware already installed. Installation by the user is only required if firmware needs to be updated.

Real Time Support (SEA 975x): This software runs on the NI RT Linux CompactRIO system. Installation is required by the user.



It is recommended to install both driver packages (Real Time and FPGA) to use the full functionality set, however, each package can be installed and operated independently from the other.

For details on the installation procedure refer to the chapter *Installation* later in this manual.

This manual covers the driver software package for the Real Time target. For information on the other software packages please refer to the respective documentation.

4 Installation

There is a single driver software for the module SEA 9741/9744/9745. The driver software can be directly downloaded and installed through the JKI VI Package Manager. Additionally the driver software can be downloaded either from the NI Tools Network or from S.E.A. download site and installed separately.



Due to continuous improvements the required software is not enclosed with the module hardware when shipped. The latest version of the driver software can be downloaded from the S.E.A. web site:

<http://www.sea-gmbh.com>

On the main page select *Support* from the main menu. On the support page select *Downloads* from the local menu and click on the *Download Area* button. On the download site select your module type from the *Hardware Products* category.

The driver software is to be installed on a development PC only (refer to the hardware requirement section below). In order to install the software package double-click or open the .vip file inside the VI Package Manager and follow the instructions on the screen. This procedure installs the driver including application programming interface (API), tools, examples and all related documentation. Further resources (if available) like application notes, drawings etc. can be downloaded separately from the location as stated above.

4.1 Hardware Requirements

The SEA 974x module requires at least a PC to program an application. In order to run the application a NI CompactRIO Real-Time controller with the SEA module inserted in any of the chassis slots is required.



The driver software requires a PC for programming (development PC) and a NI CompactRIO with FPGA chassis to run the application (run-time device).

4.2 Software Requirements

The development PC and the run-time device are described in the hardware requirements section above.

Software requirements for the development PC:

- JKI VI Package Manager 2014 or higher
- NI LabVIEW Development environment 2013 or higher
- NI LabVIEW Real-Time 2013 or higher
- NI LabVIEW FPGA 2013 or higher

Software requirements for the run-time device (NI CompactRIO):

- NI LabVIEW Run-Time Engine / LabVIEW Run-Time Engine for Real-Time targets



SEA 974x driver software does not support the *Scan Interface* mode at present.

5 Quick Start

This section describes how easy the SEA 974x module can be integrated into an existing CompactRIO system. Further ready-to-use examples are available in the example finder (keyword: 974).



Refer to the hardware manual for proper installation of the hardware.

Please follow the steps below to create a simple RT application using the SEA 974x (for this tutorial exemplary the SEA 9744 has been used):

1. Connect the SEA 974x with your CompactRIO system using the enclosed Ethernet cable. Connect it to the *Ethernet* port on the modules side. Insert the SIM card in the slot at the back of the module. Connect the antennas and place them outside of your building or next to a window. Provide power through the front side connectors. It is not required to insert the SEA 974x module in any slot for this tutorial.
2. Connect your CompactRIO system to the development PC via an Ethernet cable. Use a switch if needed. Power up the system.
3. Start LabVIEW and select *New* → *Empty Project* within the *Getting Started* window.
4. The project explorer window appears. In this window select the uppermost item in the tree (*Project: Untitled Project X.lvproj*) and select *New* → *Targets and Devices* right-clicking on it, like shown below:

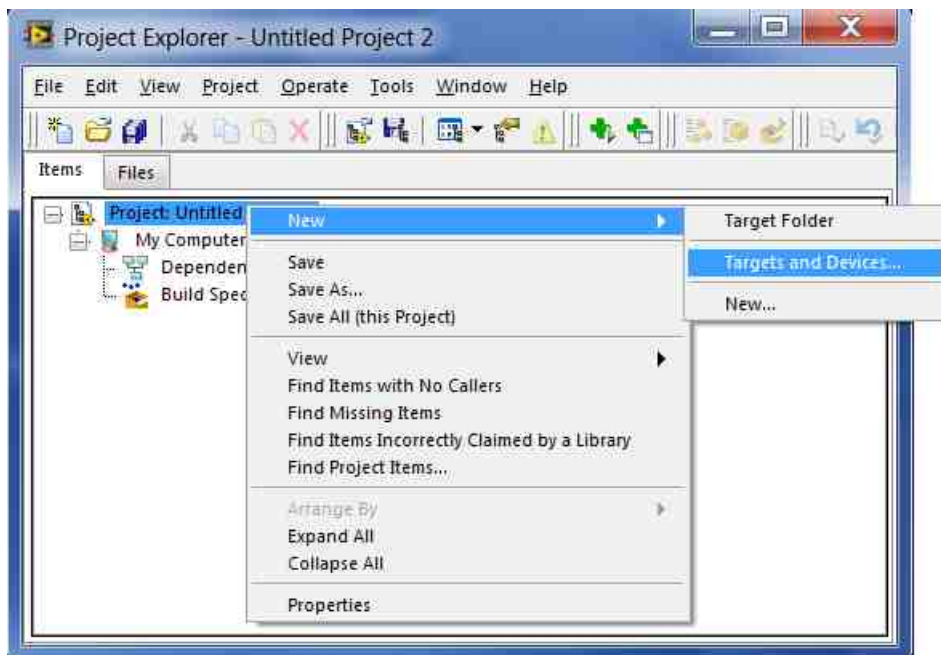


Fig. 1: Quick Start – Add Target

5. Choose *Discover an existing target(s) or device(s)* and unfold the *Real-Time CompactRIO* folder.
6. Select your CompactRIO system from the list of available targets.
7. Finalize the configuration of the FPGA, however the FPGA part is not relevant for this tutorial.

Now the project is fully configured to use the SEA 974x module in the Real Time controller (without FPGA).

You can continue from here to learn how to implement a simple Real Time application using the SEA 974x module.

8. Create a new RT VI in the project explorer window. For this right-click on the RT Target (*RIO0...*) and select *New* → *VI*. Refer to the figure below:

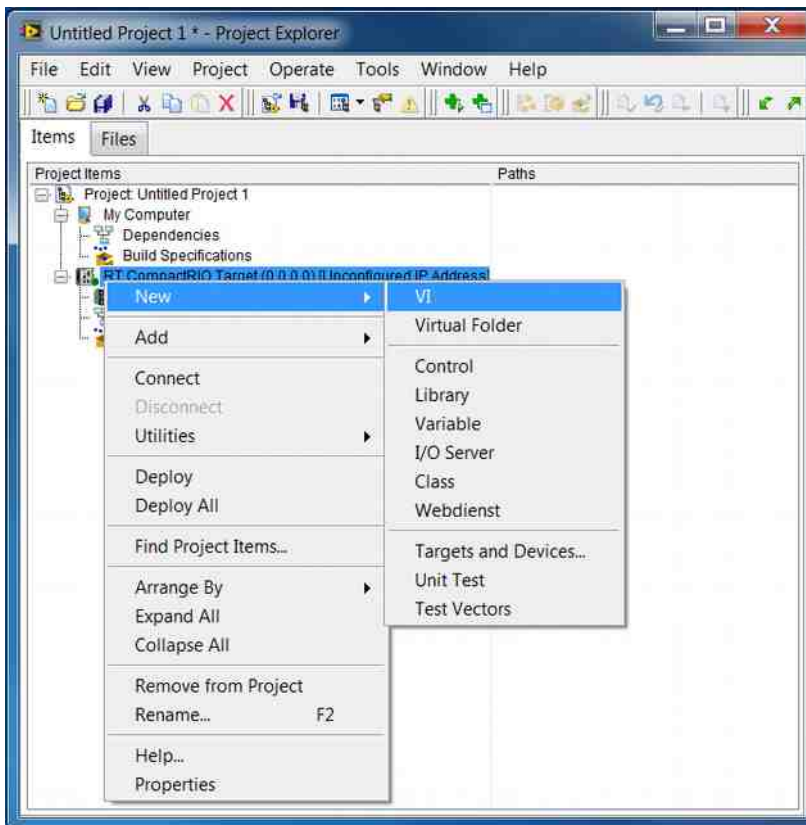


Fig. 2: Quick Start – Add VI

9. Open the block diagram of the VI you just created and place the *Open Connection VI* from the functions palette (Addons → SEA 974x / 975x) on the block diagram. Pass '192.168.1.100' to the *Module IPAddress* input. Create an indicator for the *Module Type* output.
10. Place the *Close Connection VI* on the block diagram and connect the *ModRef Out* output of the *Open Connection VI* with the *ModRef In* input of the *Close Connection VI*. Connect the *Error Out / In* connectors the same way. The Final VI should look like this:

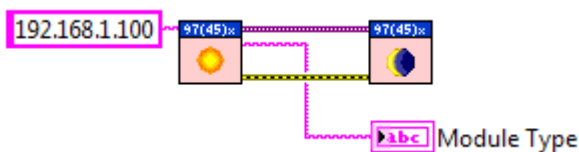


Fig. 3: Quick Start – Add VI

11. Save the created Real Time target VI and run it.
12. The Module Type '9744' should be returned, upon a successful execution.

All RealTime functions are accessible through the functions palette. For a complete list of available function, please refer to section 8.3 Application Programming Interface API.

6 Description of Functionality

6.1 CompactRIO Platform

The SEA 974x modules are compatible with all CompactRIO and CompactDAQ chassis from National Instruments. The module offers the full range of functionality through two interfaces Real Time and FPGA. In order to use the Real Time interface the module need to be connected with the CompactRIO system by the Ethernet Uplink cable. In order to use the FPGA interface the module need to be inserted in any of the slots of the CompactRIO chassis. Both interfaces can be operated independently. For each interface a separate driver software package is available.

With CompactRIO systems the full range of functionality is available, including Real Time and FPGA. With CompactDAQ systems only the Real Time interface can be used. The FPGA interface is not available as CompactDAQ systems do not provide the FPGA capability.

6.2 SEA 974x Functionality

SEA 974x features two hardware interfaces that directly interact with each of the two targets of a CompactRIO system: Real Time and FPGA.

The hardware interface to the FPGA is provided through the 15 D-SUB connector on the back side of the module. This interface allows to read GPS data (position, time, status..) directly from the SEA 974x module to the FPGA and implement very precise timings based on these data. Additionally the module's meta data, e.g. it's serial number, module ID, can be retrieved through the FPGA. GSM related functions are not available through the FPGA interface.

The hardware interface to the Real Time Controller is provided via the Uplink port of the module (Ethernet). This interface provides access to both, GPS and GSM functions, but has a higher latency when processing data than the FPGA interface.



For a full reference of all available functions using the Real Time Controller or FPGA interface please refer to the chapter *Programming* of the respective software driver manual.



The driver software packages (Real Time and FPGA) are required for operation of the SEA 974x module. They provide the application programming interfaces (API), examples and manuals. The drivers are not enclosed with the module when shipped, but need rather to be downloaded separately from the download area at www.sea-gmbh.com/support.

The following sub sections describe the features of SEA 974x in more details and explain shortly the technologies the user can take advantage of using SEA 974x. Principally the module's functionality can be divided into two main areas: GSM (mobile communication) and GPS (geo-positioning and accurate timing).

6.2.1 GSM

SEA 974x module contains a cellular engine, which allows you to establish a mobile connection with the module to access the CompactRIO system and transfer data through the mobile network. SEA 974x uses the GSM (Global System for Mobile Communications) mobile network, which is a global standard for wireless, mobile communication primary used by mobile phones.

GSM is the standard of the so called second Generation (2G) of mobile communication. With UMTS the third generation (3G) of mobile communication standard is available. HSPA even exceeds the data transfer rates of UMTS and belongs to the so called 3G+ generation of mobile communication. The latest generation (4G) represents the LTE technology which extends the limits of data transfer rates even more further. LTE is currently the fastest available technology for mobile data communication.



The term *GSM*, in fact, denotes the second generation of mobile networks (2G) but in scope of this manual it is used as a synonym for all supported mobile networks like *GSM*, *UMTS* or *LTE* unless otherwise specified. Alternately to GSM the term *mobile* is used synonymously.

GSM provides multiple services that can be used for communication and transfer data. SEA 974x module features two services for data transfer:

- *Short Message Service* (SMS)

- *Packet Data Service* (GPRS, UMTS, LTE)

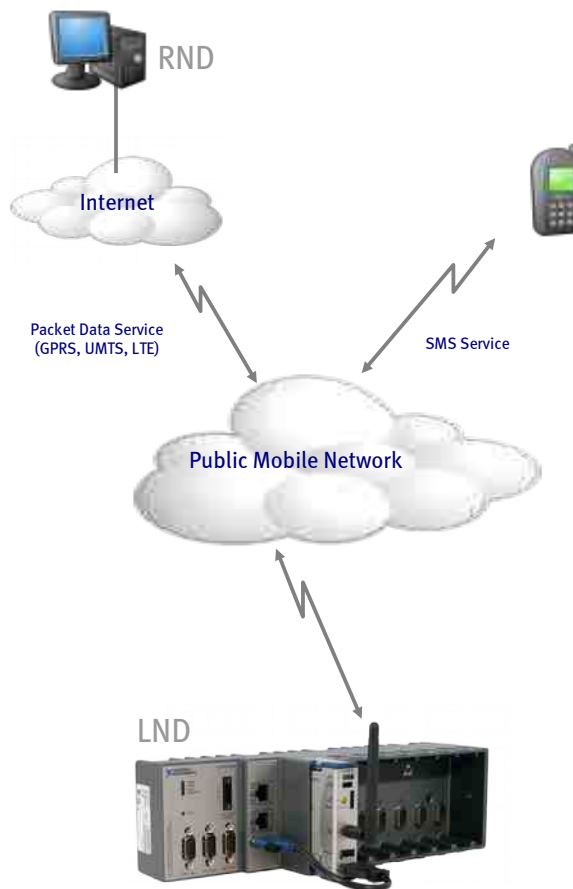


Fig. 4: GSM Overview



The access to particular services require an adequate authorization for the most MNOs. This can be a SIM card + PIN and/or Access Point name inclusive user name and password.

These settings can only be obtained form the MNO used. is not a MNO and therefore does not provide these information.

For the SMS service the driver software provides a LabVIEW API, that can be used for send and receive SMS messages within an application.

For the Packet Data Service the driver software only provides API functions to control the mobile network interface: establish, close and monitor a connection to the Internet. Once established, this connection is transparent, which means that the user application can use native TCP LabVIEW functions like TCP, FTP, Web-DAV and SMTP. The configuration of the internet connections (air interface through SEA 974x or cable connection through the real time controller) is done using a web configuration interface. For more details on the configuration refer to chapter *Configuration*.

Voice calls are not supported with the SEA 974x module. So the SEA 974x module can be described as wireless modem for mobile networks.

6.2.2 Remote Access

Remote Access refers to any incoming connection from a RND through the mobile network to the SEA 974x module. This technique is based on the GSM and allows to access LND (typically CompactRIO) for the internet (RND). As the SEA 974x acts as a network router for the CompactRIO (LND). The connection has to be made to the *External IP Address* of the SEA 974x module, which then routes the connection to the LND. Two techniques can be used: Port Forwarding or Virtual Private Network (VPN). A comparison of features provided by those techniques is made in Tab. 2.

Feature	VPN	Port Forwarding
UDP supported	yes	no
Extra Software on Client needed	yes	no
Encrypted connection	yes	no
Can be used to have two RT Systems to communicate with each other	no	yes
Easy setup	no	yes
Ports used have to be known	no	yes
Ports accessible from internet w/o extra authentication (may be a security risk)	no	yes
Allows communication of NI MAX with the RT System	yes	no

Tab. 2: Comparison of VPN an Port Forwarding

In the most cases the MNO does not provide a static external IP address, but rather assigns an IP address dynamically to the mobile network interface every time the module dials-up, which may make the usage of the IP address quite awkward. Therefore the SEA 974x comes with an option to use a dynamic domain name service (dynamic DNS), which allows to access the SEA 974x through a (static) symbolic name instead of its (dynamic) IP address.



Make sure that your MNO does assign a public routed IP address to the module, and does not block the intended ports! For VPN the port 8443 is used per default.

If the assigned IP address is in the following ranges, then it is a so-called private address and no access from the internet to the SEA 974x module is possible: 10.x.x.x, 172.16.x.x to 172.31.x.x 192.168.x.x. (However, connections from module to RND are still possible.)

The dynamic DNS as well as the access techniques are describes in more details in the following chapters. The remote access including all related technologies can be configured using the Web Configuration.



For a detailed discussion on the remote access incl. exemplary configuration refer to the application note: *SEA 974x_ApplicationNote_RemoteAccess_VPN.pdf*

6.2.2.1 Dynamic DNS

A dynamic DNS is a proprietary internet service (can be free or commercial), which converts (look up) the server's (static) symbolic name into its (dynamic) IP address. Clients can use the static symbolic name of the server rather than its dynamic (and not always know) IP address. If dynamic DNS is enabled and configured the SEA 974x module issues an http:// (or https://) request with an URL in which a symbolic name and the current IP address are encoded in order to update its IP address.

A free service is for example NoIP (noip.com). A commercial (charged) service is for example provided by DynDNS (<http://www.dyndns.com>). Please refer to your dynamic DNS service provider for details of the required URL syntax while configuring the VPN in the Web Configuration.

6.2.2.2 Port Forwarding

Any connection arriving from the internet on one of the specified ports (refer to Web Configuration) is transparently forwarded to the LND, as if the connection was directly made to it. This applies only to TCP based connections, and not for UDP based connections. Furthermore, port forwarding only applies to incoming connections. On outgoing connections all ports are open and cannot be defined or firewalled.



If VPN is enabled do not forward the port that is used by OpenVPN (def. 8443) as well as the used VPN server port (def. 1194), as this causes protocol conflicts and leads to failures at remote access.

6.2.2.3 VPN

A *Virtual Private Network* (VPN) is a computer network that uses a public telecommunication infrastructure such as the Internet to provide remote offices or individual users with secure access to their organization's network. It aims to avoid an expensive system of owned or leased lines that can be used by only one organization. It encapsulates data transfers between two or more networked devices which are not on the same private network, so as to keep the transferred data private from other devices on one or more intervening local or wide area networks. There are many different classifications, implementations, and uses for VPNs¹.

VPN allows point-to-point connections based on a Client-Server architecture. In this scenario the SEA 974x module acts as a VPN server, to which a remote machine (VPN client) can connect. VPN places the remote machine virtually inside the network behind the SEA 974x module, allowing complete network access to every network client, that is connected to the module, as if they were directly connected to the remote machine.

In Figure 5, you can see a simplified schematic overview of a VPN tunnel as used by the SEA 974x module. This connection is secured with a 2048 bit pre-shared key file. 'Pre-shared key' means that both the client and server use the same key, as opposed to a public/private key infrastructure.

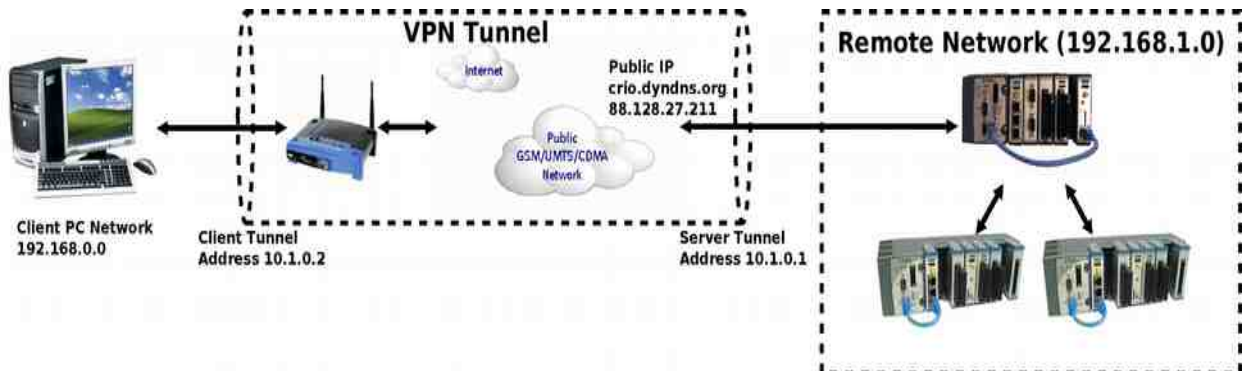


Fig. 5: VPN Schematic

As implementation on the SEA 974x modules side (server) OpenVPN has been used. For the client side the OpenVPN client should be used. The VPN client software is not included with the driver software, but need rather be downloaded from the OpenVPN web site (<http://openvpn.net>), and installed on the remote machine that aims to connect to the SEA 974x module. Please note, that the SEA 974x modules do not offer all the options and modes supported by OpenVPN. For specific needs, for example management of large numbers of remote devices please contact S.E.A. Datentechnik GmbH (info@sea-gmbh.com).



To make use of the VPN functions of the SEA 974x, you need to install the OpenVPN software available from <http://openvpn.net>. The 'Community'-version is available for download as free Open-Source-Software. Please refer to the website for installation instructions. The software is available for Windows, Mac and Linux. On the Macintosh, 'tunnelblick' is a popular OpenVPN client. It can be obtained for free from <http://code.google.com/p/tunnelblick/>.

Establishing an VPN connection can take up to some minutes.

6.2.3 GPS

The SEA 974x modules feature the reception of GPS Data (global position information and precise time information) and provide that data to be used directly within the FPGA or as a stream within the RT target.

The SEA 974x module is suitable for a wide variety of applications, including geo-position monitoring, absolute timing and synchronization.

The SEA 974x modules offer:

¹ from Wikipedia <http://en.wikipedia.org/wiki/VPN>

- Direct access to geo-positional and timing data in FPGA
- Full access to the GPS data stream in RT
- Configuration for geo-positional and timing data in FPGA
- A configurable time pulse signal in FPGA
- Customizable Sync-Output in FPGA
- Up to 4 (10) Hz message update rate
- GPS enhancements: SBAS (WAAS, EGNOS, MSAS)

All functions are accessible as nodes within the FPGA programming level or through the functions palette for the RT programming level. Only one function can be executed at a time, which means that individual functions can only be executed subsequently.



The user has to ensure that two functions are not executed at the same time, as this may lead to unexpected behavior or to malfunction.

6.2.4 Power up, Reset, Recover...

At power up, a hardware reset of the SEA 974x module is executed. The GPS receiver is configured and can be used without any further initialization.

7 Configuration

The SEA 974x features a web based configuration, which is available with a web browser connecting to the URL:



`https://[Modules IP Address]:8443`

where [Modules IP Address] is a place holder, that is to be replaced with the particular modules IP address. The default IP address of the module is 192.168.1.100, if a factory reset is executed it also gets set back to that address.

In order to perform the configuration the SEA 974x module needs to be powered on and connected to your PC through the Uplink (Ethernet) port.

The Web Configurator is considered to provide a central access to the most functions of the SEA 974x, allowing configuration, Status monitoring and debugging. The information about the current module type and its firmware version is also shown.

Access to this page is secured through SSL as well through an authentication mechanism (user name, password). For the first access the SSL certificate has to be accepted to gain access to the configuration pages.

The settings are grouped into logical sections, which are described here shortly. The detailed description of every single setting is provided as a tip text presented in the browser when moving the mouse over a particular setting.

7.1 Management

In the section the entire configuration can be transferred forth and back between the module and your computer using a configuration file. This is useful if a backup of the current configuration should be created or a particular configuration should be distributed throughout multiple modules.

The overview of available settings (configuration file) is given in the list below:

Name	Description
DynDNS_Enabled	Enables (True) or disables (false) the Dynamic DNS service
DynDNS_Hostname	Defines the host name of the used dynamic DNS service
DynDNS_UserName	User name for dynamic DNS service account
DynDNS_Password	Password for dynamic DNS service account
DynDNS_URL	URL for dynamic IP address updates (example for dynDNS: <code>https://members.-dyndns.org/nic/update?hostname={HOST}&myip={IPADDR}</code>)
GPS_BAUDRATE	Baudrate of the GPS receiver (def. 115200)
HTTP_Port	Web Configuration server port
PPP_APN	Name of the access point for packed data services in mobile network; value is provided by the MNO
PPP_OnDemand	Defines if packed data connection should be established at start-up (true) or not (false). If set to false the packet data connection need to be established either manually (per web configuration page) or programmatically (LabVIEW RT API).
PPP_Password	Password for APN; value is provided by the MNO
PPP_UserName	User name for APN; value is provided by the MNO
RemoteAccess_Port-List	List of IP ports to be forwarded to the CompactRIO real time controller
VPN_ClientIP	IP address of the VPN client
VPN_Enabled	Enables (true) or disables (false) the virtual private network (VPN).
VPN_Hostname	The host name to access the dynamic DNS service

VPN_NetworkAddr	IP address of the network behind the tunnel
VPN_NetworkNet-mask	Mask for VPN_NetworkAddr
VPN_ServerIP	IP address of the VPN server
VPN_ServerPort	IP port of the VPN server
VPN_StaticKey	Authorization key for a VPN connection

Tab. 3: Configuration File Settings

The settings in the configuration file correspond to the available on the configuration pages. Please note that names for the same setting value may slightly vary between those used in the configuration file and those used on the web pages.

For debugging purposes the user can retrieve more information about the module and control the logging level for modules firmware. The created log files are not described in more details in the scope of this document, as they are considered to be send to SEA if debugging information is required.

7.2 Network

In this section the configuration of the network adapters is done. The understanding of this section is crucial to keep full control over the behavior of the OS and user application with respect to the available ways to access the internet. The SEA 974x module adds a wireless network adapter to the CompactRIO system, which forces the user to control the decision which adapter is when to use. This decision is not done by the application, but rather by the OS. The user can control this decision through the proper configuration of the network adapter settings. The most important setting in this context is the Gateway. This value decides which adapter will be used by the application for a TCP based connection. The basic rules are as follows:

- the network adapter with a valid gateway and active link is used by the application to access a remote device
- if more than one network adapters have a valid gateway setting and active link, then the primary network adapter (typ. eth0) is preferred by the OS and therefore used by the application.



In order to use the SEA 974x module as a modem (for the LND) set the module's IP address (i.e. 192.168.1.100) as **Gateway** in the local network setting of the LND.



The LND's IP address needs to be configured in the SEA 974x configuration, because that is required as destination address for the port forwarding feature.



If you are using OpenVPN ensure that the IP addresses of the LND and the RND are not located in the same sub network, as this will prevent from setting up an TCP connection.

The reason for this is that devices within the same subnet can access each other without routing. Routing, however is performed, when a networked device outside the originators subnet has to be connected. Now, the LND and the RND are never in the same subnet. Therefore, if both are configured to be in the same sub network the connection fails because routing is not applied, although it is required.

In order to be able to establish an internet connection a valid configuration of APN (incl. user name and password) is always required. These access credentials are provided by the MNO or can be found in web.

The SEA 974x offers an option to automatically establish an internet connection on start-up. This behavior can be configured within the web configurator (setting Auto-Connect or On-Demand in configuration export file) and requires an unlocked SIM card for proper operation. The SIM card can be unlocked from the web configurator. In order to unlock the PIN is required.

7.3 Dynamic DNS

This section can be used to enable an interface to an external dynamic DNS service to access the SEA 974x module through a symbolic name. In order to use this feature an account at a provider of dynamic DNS is re-

quired. Contact your provider for the proper URL and access data and enter them in the respective fields. Note: the URL can differ between providers.



In order to use the SEA 974x module's DNS server (for the LND) set the module's IP address (i.e. 192.168.1.100) as **DNS Server** in the local network setting of the LND.

7.4 Remote Accesses

This section provides configuration settings for the both methods supported: VPN and Port Forwarding.

VPN provides two sub sections: VPN Configuration and VPN Static Key. The subsection VPN Configuration configures the IP addresses for VPN. In the next section a static key can be managed.

The subsection Port Forwarding allows you to manage a list of incoming ports that gets forwarded to the LND (typically CompactRIO). If the list is empty no port are forwarded at all.

In order to access the LND through VPN the IP address of the CompactRIO system (i.e. 192.168.1.105) has to be used for a TCP connection.

7.5 Radio Status

This section is considered to manually control and monitor the modules status with respect to the GSM functionality. This is helpful for set up and debugging purposes. Furthermore the behavior of the PIN authentication can be configured.

8 Programming

8.1 Migration



The Real Time driver software version 2.0 comes with some minor improvements (when compared to the previous version), which requires some adaptation to the user application. These adaptations are fairly easy and self-explaining, nevertheless an additional document has been created to cover this issue in details. This document can be free downloaded from the S.E.A. web site:

<http://www.sea-gmbh.com>

On the main page select *Support* from the main menu. On the support page select *Downloads* from the local menu and click on the *Download Area* button. On the download site select your module type from the *Hardware Products* category.

8.2 Examples

The driver software is delivered with a set of examples that demonstrate a specific aspect of the module. Please refer to the examples to learn how to program the module and how to retrieve position and time values.



The examples are available via the `example finder`.

Use the search keyword *974* or *975* (depending on used module type) to find related examples.

8.3 Application Programming Interface API

SEA 974x features a LabVIEW API that is available from the functions palette after installation of the driver software. The functions are organized in three main groups:

- Common module related functions (rose background)
- GPS related functions (blue background)
- GSM related functions (yellow background)

Please note that some functions may be not available for particular module types. If a function is not available it returns an error with the code 5700 when it is executed. Each API function provides two common optional connectors: *Timeout (ms)* and *Duration (ms)*. The *Timeout (ms)* input is only required if the default time out is insufficient for special purposes. Normally, the default time out is sufficient for the majority of applications. The *Duration (ms)* output can be used for benchmarking or debugging the application. Both optional parameter are omitted in the subsequent chapters.

The subsequent chapters describe all API functions in more details.

8.3.1 Close Connection

Closes the communication reference to the module.

Parameter	Data Type	Direction	Description
ModRef In	Variant	input	Accepts the communication reference from the previous API VI. This reference is created in the <i>Open Communication VI</i> and need be passed to the subsequent API VIs.
Error In	LV Error Cluster	input	Accepts the error cluster from the previous API VI. This function executes regardless the passed status.
Error Out	LV Error Cluster	output	Returns the current error status to be passed to the subsequent VI.

Tab. 4: API - Close Connection

8.3.2 Get Configuration

Returns the current module configuration. The configuration is returned as name=value pairs.

Parameter	Data Type	Direction	Description
ModRef In	Variant	input	Accepts the communication reference from the previous API VI. This reference is created in the <i>Open Communication</i> VI and need be passed to the subsequent API VIs.
Error In	LV Error Cluster	input	Accepts the error cluster from the previous API VI. This function executes regardless the passed status.
ModRef Out	Variant	output	Returns the communication reference to be passed to the subsequent API VI.
Module Configuration	String	output	Returns the current module configuration as name=value pairs.
Error Out	LV Error Cluster	output	Returns the current error status to be passed to the subsequent VI.

Tab. 5: API - Get Configuration

8.3.3 Get Version

Retrieves the current hardware and firmware versions of the module as well as the initial firmware version when being shipped. If one of the versions is not available N/A is returned instead, for details please refer to the VI description.

Parameter	Data Type	Direction	Description
ModRef In	Variant	input	Accepts the communication reference from the previous API VI. This reference is created in the <i>Open Communication</i> VI and need be passed to the subsequent API VIs.
Error In	LV Error Cluster	input	Accepts the error cluster from the previous API VI. This function executes regardless the passed status.
Version	Cluster	output	Returns the associated version numbers of the module hardware and software.
Error Out	LV Error Cluster	output	Returns the current error status to be passed to the subsequent VI.

Tab. 6: API - Get Version

8.3.4 Open Connection

Establishes a connection to the module.

Parameter	Data Type	Direction	Description
Module IPAddress	string	input	Accepts IP address in dot notation. For SEA 975x enter 127.0.0.1, for SEA 974x enter the module's IP address (def. 192.168.1.100)
Logging Filepath	Path	input	Empty Path: Logging disabled Valid abs. Filepath: location of the file where the logging data is written to.
Error In	LV Error Cluster	input	Accepts the error cluster from the previous API VI. This function executes regardless the passed status.
ModRef Out	Variant	output	Returns the communication reference to be passed to the subsequent API VI.
Module Type	String	output	Returns the module type, e.g. '9744' or '9754'
Error Out	LV Error Cluster	output	Returns the current error status to be passed to the subsequent VI.

Tab. 7: API - Open Connection

8.3.5 GPS - Enable/Disable Messages

Enables or disables the reception of the denoted NMEA-0183 message.

Parameter	Data Type	Direction	Description
ModRef In	Variant	input	Accepts the communication reference from the previous API VI. This reference is created in the <i>Open Communication</i> VI and need be passed to the subsequent API VIs.
Message	Enum	input	NMEA message to be enabled or disabled.
Enable=T (Disable=F)	Boolean	input	Enables (T) or disables (F) the selected NMEA message.
Error In	LV Error Cluster	input	Accepts the error cluster from the previous API VI. This function executes regardless the passed status.
ModRef Out	Variant	output	Returns the communication reference to be passed to the subsequent API VI.
Error Out	LV Error Cluster	output	Returns the current error status to be passed to the subsequent VI.

Tab. 8: API - GPS - Enable/Disable Messages

8.3.6 GPS - Enable/Disable SBAS

Enables or disables the Satellite Based Augmentation System (SBAS) of the GPS receiver.

Parameter	Data Type	Direction	Description
ModRef In	Variant	input	Accepts the communication reference from the previous API VI. This reference is created in the <i>Open Communication</i> VI and need be passed to the subsequent API VIs.
Enable=T (Disable=F)	Boolean	input	Enables (T) or disables (F) the SBAS functionality.
Error In	LV Error Cluster	input	Accepts the error cluster from the previous API VI. This function executes regardless the passed status.
ModRef Out	Variant	output	Returns the communication reference to be passed to the subsequent API VI.
Error Out	LV Error Cluster	output	Returns the current error status to be passed to the subsequent VI.

Tab. 9: API - GPS - Enable/Disable SBAS

8.3.7 GPS - Read Messages

Reads data from the GPS module.

Parameter	Data Type	Direction	Description
ModRef In	VARIANT	input	Accepts the communication reference from the previous API VI. This reference is created in the <i>Open Communication</i> VI and need be passed to the subsequent API VIs.
Clear Buffer (F)	Boolean	input	If true, the internal message buffer is cleared.
Error In	LV Error Cluster	input	Accepts the error cluster from the previous API VI. This function executes regardless the passed status.
ModRef Out	VARIANT	output	Returns the communication reference to be passed to the subsequent API VI.
GPS Messages	Cluster	output	Cluster of GPS messages grouped by format. Only the last message of each type is returned.
GPS Status	Cluster	output	Status cluster of the GPS module.
Error Out	LV Error Cluster	output	Returns the current error status to be passed to the subsequent VI.

Tab. 10: API - GPS - Read Messages

8.3.8 GPS - Reset GPS

Performs a software reset of the GPS receiver according to the parameter Type.

Parameter	Data Type	Direction	Description
ModRef In	VARIANT	input	Accepts the communication reference from the previous API VI. This reference is created in the <i>Open Communication</i> VI and need be passed to the subsequent API VIs.
Type	Enum	input	Defines the type of reset: <i>Hotstart</i> : restarts the receiver without clearing any internal data. <i>Warmstart</i> : restarts the receiver and clears the ephemeris data. <i>Coldstart</i> : restarts the receiver and clears all data (ephemeris, almanach, health, klobuchard, position...) The Method also resets custom settings like the selected messages or update rate. So ensure to re-run the Configure Method nodes after executing a reset.
Error In	LV Error Cluster	input	Accepts the error cluster from the previous API VI. This function executes regardless the passed status.
ModRef Out	VARIANT	output	Returns the communication reference to be passed to the subsequent API VI.
Error Out	LV Error Cluster	output	Returns the current error status to be passed to the subsequent VI.

Tab. 11: API - GPS - Reset GPS

8.3.9 GPS - Set Timepulse Interval

Sets the period (time between two rising edges) of the Timepulse signal.

Parameter	Data Type	Direction	Description
ModRef In	Variant	input	Accepts the communication reference from the previous API VI. This reference is created in the <i>Open Communication</i> VI and need be passed to the subsequent API VIs.
Timepulse Interval (μ s)	U32	input	Defines the pulse repetition time (period).
Error In	LV Error Cluster	input	Accepts the error cluster from the previous API VI. This function executes regardless the passed status.
ModRef Out	Variant	output	Returns the communication reference to be passed to the subsequent API VI.
Error Out	LV Error Cluster	output	Returns the current error status to be passed to the subsequent VI.

Tab. 12: API - GPS - Set Timepulse Interval

8.3.10 GSM – Init GSM

Checks if the module is already registered with a network and registers if not yet done.

Parameter	Data Type	Direction	Description
ModRef In	Variant	input	Accepts the communication reference from the previous API VI. This reference is created in the <i>Open Communication</i> VI and need be passed to the subsequent API VIs.
PIN/PUK	String	input	PIN is a four digit number of used SIM card, or PUK an eight digit number.
Error In	LV Error Cluster	input	Accepts the error cluster from the previous API VI. This function executes regardless the passed status.
ModRef Out	Variant	output	Returns the communication reference to be passed to the subsequent API VI.
Registered	Boolean	output	Indicates of the module is registered on network (T) or not (F).
Roaming	Boolean	output	Indicates if the module is registered in roaming mode (T) or not (F).
Error Out	LV Error Cluster	output	Returns the current error status to be passed to the subsequent VI.

Tab. 13: API - GSM - Init GSM

8.3.11 GSM – Query IMEI

Queries the International Mobile Equipment Identity (IMEI) of the module.

Parameter	Data Type	Direction	Description
ModRef In	Variant	input	Accepts the communication reference from the previous API VI. This reference is created in the <i>Open Communication</i> VI and need be passed to the subsequent API VIs.
Error In	LV Error Cluster	input	Accepts the error cluster from the previous API VI. This function executes regardless the passed status.
ModRef Out	Variant	output	Returns the communication reference to be passed to the subsequent API VI.
IMEI Code	String	output	Returns the International Mobile Equipment Identity of the module. This is a module's unique identification number.
Error Out	LV Error Cluster	output	Returns the current error status to be passed to the subsequent VI.

Tab. 14: API - GSM – Query IMEI

8.3.12 GSM – Query Network

Queries the network settings the module is registered to.

Parameter	Data Type	Direction	Description
ModRef In	Variant	input	Accepts the communication reference from the previous API VI. This reference is created in the <i>Open Communication</i> VI and need be passed to the subsequent API VIs.
Error In	LV Error Cluster	input	Accepts the error cluster from the previous API VI. This function executes regardless the passed status.
ModRef Out	Variant	output	Returns the communication reference to be passed to the subsequent API VI.
Network	Cluster	output	Returns information about currently used network.
Error Out	LV Error Cluster	output	Returns the current error status to be passed to the subsequent VI.

Tab. 15: API - GSM - Query Network

8.3.13 GSM – Query Registration

Queries if the module is registered to the network and returns the roaming status.

Parameter	Data Type	Direction	Description
ModRef In	Variant	input	Accepts the communication reference from the previous API VI. This reference is created in the <i>Open Communication</i> VI and need be passed to the subsequent API VIs.
Error In	LV Error Cluster	input	Accepts the error cluster from the previous API VI. This function executes regardless the passed status.
ModRef Out	Variant	output	Returns the communication reference to be passed to the subsequent API VI.
Registered	Boolean	output	Indicates if the module is registered (T) on network or not (F).
Roaming	Boolean	output	Indicates is the module is registered in roaming mode (T) or not (F).
Error Out	LV Error Cluster	output	Returns the current error status to be passed to the subsequent VI.

Tab. 16: API - GSM - Query Registration

8.3.14 GSM – Query Service Status

Queries the mobile service status information like availability, domain, roaming or SIM status.

Parameter	Data Type	Direction	Description
ModRef In	Variant	input	Accepts the communication reference from the previous API VI. This reference is created in the <i>Open Communication</i> VI and need be passed to the subsequent API VIs.
Error In	LV Error Cluster	input	Accepts the error cluster from the previous API VI. This function executes regardless the passed status.
ModRef Out	Variant	output	Returns the communication reference to be passed to the subsequent API VI.
Service Status	Cluster	output	Service Status Information
Error Out	LV Error Cluster	output	Returns the current error status to be passed to the subsequent VI.

Tab. 17: API - GSM - Query Service Status

8.3.15 GSM – Query Signal Quality

Queries the signal quality that the module is currently receiving.

Parameter	Data Type	Direction	Description																				
ModRef In	Variant	input	Accepts the communication reference from the previous API VI. This reference is created in the <i>Open Communication</i> VI and need be passed to the subsequent API VIs.																				
Error In	LV Error Cluster	input	Accepts the error cluster from the previous API VI. This function executes regardless the passed status.																				
ModRef Out	Variant	output	Returns the communication reference to be passed to the subsequent API VI.																				
Signal Strength (RSSI)	I32	output	<p>Indicates the received signal strength indicator (RSSI). This value is in a range 0 to 31. The higher the value the better the signal strength.</p> <p>This value can be converted in dBm: $\text{dBm} = (\text{RSSI} * 2) - 113$.</p> <p>dBm denotes the signal strength in decibel. dBm=0 means that the received signal is 1 mW. The signal strength is usually lower than 1 mW therefore the dBm value is always below 0. The range is -113 dBm (RSSI=0) to -51 dBm (RSSI=31). Whereby a value of RSSI=0 also all values less than -113dBm and -51dBm all values above -51dBm includes.</p> <p>If the value is not detectable by the module 99 is returned and warning code 5726 is returned in <i>error out</i> cluster.</p>																				
Signal Quality (RXQUAL)	I32	output	<p>Indicates the received (averaged) signal quality. This value is in a range 0 to 7. The lower the value the better the signal quality (less data transfer errors).</p> <p>Some module types may not support this value. If this value is not detectable 99 is returned instead and warning code 5726 is returned in <i>error out</i> cluster.</p> <p>If the Signal Quality (RXQUAL) is detectable, the Bit Error Rate (BER) can be derived. The Bit Error Rate (BER) is an estimated value, which provides a measure about the error probability of the current signal. The conversion table is as follows:</p> <table border="1"> <thead> <tr> <th>RXQUAL</th> <th>BER</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0 to 0,2 % (good)</td> </tr> <tr> <td>1</td> <td>0,2 to 0,4 %</td> </tr> <tr> <td>2</td> <td>0,4 to 0,8 %</td> </tr> <tr> <td>3</td> <td>0,8 to 1,6 %</td> </tr> <tr> <td>4</td> <td>1,6 to 3,2 %</td> </tr> <tr> <td>5</td> <td>3,2 to 6,4 %</td> </tr> <tr> <td>6</td> <td>6,4 to 12,8 %</td> </tr> <tr> <td>7</td> <td>12,8 to 100 % (poor)</td> </tr> <tr> <td>99</td> <td>---</td> </tr> </tbody> </table>	RXQUAL	BER	0	0 to 0,2 % (good)	1	0,2 to 0,4 %	2	0,4 to 0,8 %	3	0,8 to 1,6 %	4	1,6 to 3,2 %	5	3,2 to 6,4 %	6	6,4 to 12,8 %	7	12,8 to 100 % (poor)	99	---
RXQUAL	BER																						
0	0 to 0,2 % (good)																						
1	0,2 to 0,4 %																						
2	0,4 to 0,8 %																						
3	0,8 to 1,6 %																						
4	1,6 to 3,2 %																						
5	3,2 to 6,4 %																						
6	6,4 to 12,8 %																						
7	12,8 to 100 % (poor)																						
99	---																						
Error Out	LV Error Cluster	output	Returns the current error status to be passed to the subsequent VI.																				

Tab. 18: API - GSM - Query Signal Quality

8.3.16 GSM – Reset GSM

Performs a software reset of the cellular engine, which resides on the module.

Parameter	Data Type	Direction	Description
ModRef In	Variant	input	Accepts the communication reference from the previous API VI. This reference is created in the <i>Open Communication</i> VI and need be passed to the subsequent API VIs.
Error In	LV Error Cluster	input	Accepts the error cluster from the previous API VI. This function executes regardless the passed status.
ModRef Out	Variant	output	Returns the communication reference to be passed to the subsequent API VI.
Error Out	LV Error Cluster	output	Returns the current error status to be passed to the subsequent VI.

Tab. 19: API - GSM - Reset GSM

8.3.17 SMS – Delete SMS

Deletes a single SMS message.

Parameter	Data Type	Direction	Description
ModRef In	Variant	input	Accepts the communication reference from the previous API VI. This reference is created in the <i>Open Communication</i> VI and need be passed to the subsequent API VIs.
Index ID	I32	input	SMS reference index, which is returned by the <i>GSM_SMS_Receive.vi</i> function.
Error In	LV Error Cluster	input	Accepts the error cluster from the previous API VI. This function executes regardless the passed status.
ModRef Out	Variant	output	Returns the communication reference to be passed to the subsequent API VI.
Error Out	LV Error Cluster	output	Returns the current error status to be passed to the subsequent VI.

Tab. 20: API - SMS - Delete SMS

8.3.18 SMS – Init SMS

Initializes the module for transferring SMS messages.

Parameter	Data Type	Direction	Description
ModRef In	Variant	input	Accepts the communication reference from the previous API VI. This reference is created in the <i>Open Communication</i> VI and need be passed to the subsequent API VIs.
SMS Mode	Enum	input	
MO SMS Service	Enum	input	MO SMS Service selector: Packet Domain (0) Circuit switched (1) Packet Domain Preferred (2) (Use Circuit Switched if GPRS not available) Circuit Switched Preferred (3) (Use Packet Domain if Circuit Switched not available)
Error In	LV Error Cluster	input	Accepts the error cluster from the previous API VI. This function executes regardless the passed status.

ModRef Out	VARIANT	output	Returns the communication reference to be passed to the subsequent API VI.
Error Out	LV Error Cluster	output	Returns the current error status to be passed to the subsequent VI.

Tab. 21: API - SMS - Init SMS

8.3.19 SMS – Query SMS Service Center

Returns the actual Service Center Address (SCA) and type of SCA (TOSCA).

Parameter	Data Type	Direction	Description
ModRef In	VARIANT	input	Accepts the communication reference from the previous API VI. This reference is created in the <i>Open Communication</i> VI and need be passed to the subsequent API VIs.
Error In	LV Error Cluster	input	Accepts the error cluster from the previous API VI. This function executes regardless the passed status.
ModRef Out	VARIANT	output	Returns the communication reference to be passed to the subsequent API VI.
Service Centre Address (SCA)	String	output	The SMS Service Center Address.
Type Of Service Center Address (TOSCA)	Enum	output	Either international (number begins with ""+""") or national number scheme.
Error Out	LV Error Cluster	output	Returns the current error status to be passed to the subsequent VI.

Tab. 22: API - SMS - Query SMS Service Center

8.3.20 SMS – Receive SMS

Reads SMS messages according to the filter option.

Parameter	Data Type	Direction	Description
ModRef In	VARIANT	input	Accepts the communication reference from the previous API VI. This reference is created in the <i>Open Communication</i> VI and need be passed to the subsequent API VIs.
Group	Enum	input	SMS Message Status selector: REC UNREAD: only received but not yet read messages. REC READ: received and read messages. STO UNSENT: only stored but not yet sent messages. STO SENT: only stored and already sent messages. ALL: all received and stored messages.
Error In	LV Error Cluster	input	Accepts the error cluster from the previous API VI. This function executes regardless the passed status.
ModRef Out	VARIANT	output	Returns the communication reference to be passed to the subsequent API VI.
SMS Messages	Cluster	output	Returns an array of received messages: <ul style="list-style-type: none"> • Index ID is the message reference index (position on the storage) • Stat is the status of the message. See description of the Group input parameter. • SmsFrom / Length is the sender number (text mode) or messages length (pdu mode)

			<ul style="list-style-type: none"> Message / PDU Code is the message text (text mode) or the pdu code (pdu mode)
SMS Messages (Plain)	String	output	Returns an array of SMS messages flattened into string.
Error Out	LV Error Cluster	output	Returns the current error status to be passed to the subsequent VI.

Tab. 23: API - SMS - Receive SMS

8.3.21 SMS – Send SMS

Sends a SMS to the given destination phone number.

Parameter	Data Type	Direction	Description
ModRef In	Variant	input	Accepts the communication reference from the previous API VI. This reference is created in the <i>Open Communication</i> VI and need be passed to the subsequent API VIs.
Destination Phone Number	String	input	The phone number of the SMS recipient. The phone number must apply the one of the numbering schemata: <ul style="list-style-type: none"> international number scheme (e.g. +491771222222) national number scheme (e.g. 01771222222) To check which numbering scheme your operator supports use the SMS – Query SMS Service Center VI (refer to TOSCA).
SMS Message	String	input	The message text string to be send (max 160 chars).
Error In	LV Error Cluster	input	Accepts the error cluster from the previous API VI. This function executes regardless the passed status.
ModRef Out	Variant	output	Returns the communication reference to be passed to the subsequent API VI.
Message Reference Number (MRN)	I32	output	Returns a number, which indicates the network reference for identifying the sent message.
Error Out	LV Error Cluster	output	Returns the current error status to be passed to the subsequent VI.

Tab. 24: API - SMS - Send SMS

8.3.22 Packet - Close Packet Connection

Closes the previously opened data connection to the internet and waits until the connection is destroyed.

Parameter	Data Type	Direction	Description
ModRef In	Variant	input	Accepts the communication reference from the previous API VI. This reference is created in the <i>Open Communication</i> VI and need be passed to the subsequent API VIs.
Error In	LV Error Cluster	input	Accepts the error cluster from the previous API VI. This function executes regardless the passed status.
ModRef Out	Variant	output	Returns the communication reference to be passed to the subsequent API VI.
Disconnected	Boolean	output	Indicates if the packet data connection (internet) is destroyed. T = disconnected.
Error Out	LV Error Cluster	output	Returns the current error status to be passed to the subsequent VI.

Tab. 25: API - Packet - Close Packet Connection

8.3.23 Packet – Get External IP Address

Returns the external IP address of the module.

Parameter	Data Type	Direction	Description
ModRef In	Variant	input	Accepts the communication reference from the previous API VI. This reference is created in the <i>Open Communication</i> VI and need be passed to the subsequent API VIs.
Error In	LV Error Cluster	input	Accepts the error cluster from the previous API VI. This function executes regardless the passed status.
ModRef Out	Variant	output	Returns the communication reference to be passed to the subsequent API VI.
External IP Address	String	output	Returns the external IP address, refer to VI description for details.
Error Out	LV Error Cluster	output	Returns the current error status to be passed to the subsequent VI.

Tab. 26: API - Packet - Get External IP Address

8.3.24 Packet – Open Packet Connection

Opens a data connection to the internet and waits until the connection is established.

Parameter	Data Type	Direction	Description
ModRef In	Variant	input	Accepts the communication reference from the previous API VI. This reference is created in the <i>Open Communication</i> VI and need be passed to the subsequent API VIs.
Error In	LV Error Cluster	input	Accepts the error cluster from the previous API VI. This function executes regardless the passed status.
ModRef Out	Variant	output	Returns the communication reference to be passed to the subsequent API VI.
Connected	Boolean	output	Indicates if the packet data (internet) connection is established. T = connected.
Error Out	LV Error Cluster	output	Returns the current error status to be passed to the subsequent VI.

Tab. 27: API - Packet - Open Packet Connection

8.3.25 Packet - Query Packet Connection Status

Returns the status of the connection to the internet.

Parameter	Data Type	Direction	Description
ModRef In	Variant	input	Accepts the communication reference from the previous API VI. This reference is created in the <i>Open Communication</i> VI and need be passed to the subsequent API VIs.
Error In	LV Error Cluster	input	Accepts the error cluster from the previous API VI. This function executes regardless the passed status.
ModRef Out	Variant	output	Returns the communication reference to be passed to the subsequent API VI.
Connected	Boolean	output	Indicates if the packet data (internet) connection is established.
Error Out	LV Error Cluster	output	Returns the current error status to be passed to the subsequent VI.

Tab. 28: API - Packet - Query Packet Connection Status

8.4 Error Codes

In case of unexpected behavior the driver software returns an error within the error cluster of the originating API function. The error codes are in the range 5700 to 5731. The meaning of each error code is explained in the table *Error Codes* below or can be found when selecting the *Explain Error..* menu item in LabVIEW.

The following errors and warnings are possible:

Error	Description
5700	Function not available for current module type.
5701	No or invalid module found.
5702	Module's IP address is empty or ill-formatted.
5703	Internal command to module timed out.
5704	Internal command to module returned with error response. Some input parameter may be invalid. Function may be undefined or not available in present status.
5705	Internal command to module returned with too many characters.
5706	Cellular engine related error occurred
5707	Module could not be registered on mobile network.
5708	Error while querying cellular network registration status.
5709	SMS related error occurred.
5710	SMS is not initialized.
5711	Waiting for registration on network timed out.
5712	Waiting for packed data connection timed out.
5713	Waiting for packed data disconnection timed out.
5714	Connection to the module lost.
5723	Undefined error type occurred.
Warnings	
5724	Cellular engine reports temperature warning.
5725	Warning while registering on cellular network.

5726	Warning while querying cellular network registration status.
5727	The parameter denoted above is not available for current module type (cellular engine does not provide this value).

Tab. 29: Error Codes

9 Deployment

If an application containing the SEA 974x is deployed to a run-time target system (NI CompactRIO) the following parts are necessary to be deployed along with the compiled application (i.e. rtexe):

- LabVIEW Run-Time Engine 2013 or higher

10 Trouble Shooting

The following hints might help you to determine if the module and/or software behave correctly and how to identify module failures.

#	Problem	How to solve
1	The module is not detected within LabVIEW	<ol style="list-style-type: none"> 1. Make sure you have the CompactRIO system powered up and connected to your PC. 2. Make sure you have installed the current driver software. 3. Enable the error terminals in all FPGA nodes and see if any errors occur.
2	GPS Data is not available	<ol style="list-style-type: none"> 1. Wait until LED 2 on the module blinks. The GPS receiver needs some time to collect data from at least 4 satellites until the first data is returned. 2. Check if the GPS antenna is connected to the module and placed outside buildings or next to a window. 3. Disable SBAS. 4. Enable the error terminals in all FPGA nodes and see if any errors occur
3	GPS data is not returned or some data is missing	<ol style="list-style-type: none"> 1. Disable SBAS. 2. Reduce number of messages to be returned from the GPS receiver. 3. Reduce update rate.
4	The module cannot connect to mobile network	<ol style="list-style-type: none"> 1. Check if the module is provided with adequate power through the front side connector. 2. Check if module is connected to the CompactRIO through the Ethernet uplink cable (SEA 974x) or USB uplink cable (SEA 975x). 3. Check if SIM card is present and valid. 4. Check if GSM antenna is connected and sufficient reception is available. 5. Wait until LED 3 on the module turns on. 6. SEA 974x only: Wait until LED 5 is steady on, which means 'ready' (after blinking, which means 'booting')
5	The module cannot establish a packet data connection	<ol style="list-style-type: none"> 1. Check all solution suggestions from issue #4 2. Check the APN settings incl. User name and password. 3. Check with your MNO if packet data is enabled with the SIM card used. 4. Check if sufficient funds are available for the used SIM card (pre-paid). 5. Wait until LED 4 on the module turns on.
6	No connection to the module is possible	<ol style="list-style-type: none"> 1. Check that the right IP address of your module is used for the MOD Open API VI. 2. Check the Ethernet (SEA 974x) or USB (SEA 975x) connection between the module and the CompactRIO. 3. Ensure that the IP addresses of your module and the CompactRIO are in the same subnet.
7	VPN Connection to module not possible	<ol style="list-style-type: none"> 1. Check the internet connection (e.g. ping). 2. Check VPN settings on the module's side.

		<ol style="list-style-type: none">3. Do <u>not</u> forward the port that is used by OpenVPN (def. 8443) as well as the used VPN server port (def. 1194).4. Ensure that the local (private) IP address range on the module's side differs from the range on the PC's side. Same settings like 192.168.o.x/24 on both ends are not allowed. Different settings like 192.168.o.x/24 (module) and 192.168.1. x/24 (PC) are allowed.
--	--	--

Tab. 30: Trouble Shooting

A Figures

Fig. 1: Quick Start – Add Target.....	10
Fig. 2: Quick Start – Add VI.....	11
Fig. 3: Quick Start – Add VI.....	11
Fig. 4: GSM Overview.....	13
Fig. 5: VPN Schematic.....	15

B Tables

Tab. 1: Nomenclature.....	7
Tab. 2: Comparison of VPN an Port Forwarding.....	14
Tab. 3: Configuration File Settings.....	18
Tab. 4: API - Close Connection.....	20
Tab. 5: API - Get Configuration.....	21
Tab. 6: API - Get Version.....	21
Tab. 7: API - Open Connection.....	21
Tab. 8: API - GPS - Enable/Disable Messages.....	22
Tab. 9: API - GPS - Enable/Disable SBAS.....	22
Tab. 10: API - GPS - Read Messages.....	23
Tab. 11: API - GPS - Reset GPS.....	23
Tab. 12: API - GPS - Set Timepulse Interval.....	24
Tab. 13: API - GSM - Init GSM.....	24
Tab. 14: API - GSM – Query IMEI.....	25
Tab. 15: API - GSM - Query Network.....	25
Tab. 16: API - GSM - Query Registration.....	25
Tab. 17: API - GSM - Query Service Status.....	26
Tab. 18: API - GSM - Query Signal Quality.....	27
Tab. 19: API - GSM - Reset GSM.....	28
Tab. 20: API - SMS - Delete SMS.....	28
Tab. 21: API - SMS - Init SMS.....	29
Tab. 22: API - SMS - Query SMS Service Center.....	29
Tab. 23: API - SMS - Receive SMS.....	30
Tab. 24: API - SMS - Send SMS.....	30
Tab. 25: API - Packet - Close Packet Connection.....	30
Tab. 26: API - Packet - Get External IP Address.....	31
Tab. 27: API - Packet - Open Packet Connection.....	31
Tab. 28: API - Packet - Query Packet Connection Status.....	32
Tab. 29: Error Codes.....	33
Tab. 30: Trouble Shooting.....	36