# STAR-System QNX

STAR-System QNX is the API and driver system provided with STAR-Dundee's new range of SpaceWire devices, for the QNX operating system. STAR-System presents a common interface across operating systems, and allows different types of SpaceWire devices to be accessed in a consistent manner.

## Application Programming Interface

A full API is provided to expose all functions of supported devices allowing them to be controlled from application software. The API is provided in C, which also supports C++ applications and the API to be imported into other languages. (Check website for details of current and planned language support.)

The API is common across all of STAR-Dundee's new products, and is consistent for each programming language and platform supported. This simplifies software development and allows migration of test software from one device to another and from one platform to another, enhancing software reuse and reducing the risk of schedule delays. The API also takes advantage of features provided by each language, making the API as efficient and easy to use as possible.

A key feature of the API is that it not only provides functionality to transmit and receive packets, but also includes functions required when testing equipment. For example, the API makes it simple to transmit packets terminated with an EEP, and to determine the end of packet marker type of received packets. It simplifies the process of transmitting a stream of traffic (for example, from a file), and receiving a stream of traffic. If these streams were to include time-codes in the middle of packets, the position of these time-codes would be maintained by the API. This allows traffic streams to be recorded accurately and replayed, so multiple test runs will be consistent.

STAR-System is not only feature rich, but delivers these features with very high performance and low CPU usage. The graph below shows data when transmitting and receiving packets of various sizes on a 200 Mbits/s link. With all packet sizes ranging from as small as 50 bytes in length, the data rate achieved is close to the theoretical maximum slightly less than 160 Mbits/s.
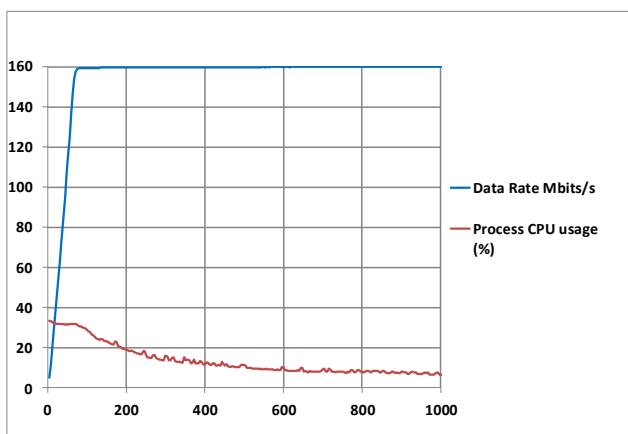


The API is provided with extensive searchable documentation and example code for each API function. Example test programs with source code are included for each language supported.

## Supporting Modules

STAR-System is a modular system. This reduces the footprint of the system, which is important where memory or storage is at a premium. If validation of the API is required, this also reduces the number of functions which must be tested.

In addition to the core components of the system, modules are provided to implement commonly required features. For users working with the Remote Memory Access Protocol, the RMAP Packet Library simplifies building and interpreting RMAP packets.

Configuration APIs are provided to configure devices supported by STAR-System. As well as allowing the configuration of devices connected to the local PC via PCI and USB buses,, these APIs also allow supported devices to be configured over a SpaceWire network. The Router Configuration API provides functions to configure features common to a number of devices, including starting and stopping links, configuring routing tables and reading the status of devices.

The Mk2 Configuration API provides functions for configuring features specific to the Mk2 range of devices, such as getting and setting the link speed, enabling and disabling interface mode and enabling the device as a time-code master. A PCI Mk2 Configuration API provides additional functions for getting and setting the link speed of PCI and cPCI Mk2 devices, which have slightly different options for the link speed.

The Router Configuration API can also be used to remotely configure STAR-Dundee devices not currently supported directly by STAR-System, such as the SpaceWire-USB Brick, SpaceWire Router-USB, SpaceWire Router Mk2 and the ESA SpW-10X.

## GUI Applications

In addition to the extensive examples, Graphical User Interface (GUI) applications are provided to perform common tasks.

**Device Configuration GUI:** Provides an interface (shown below) in which the properties of a device can be viewed and configured. From this application the device can be reset, its version can be viewed, links can be started and stopped, routing tables can be configured and many other operations can be performed.
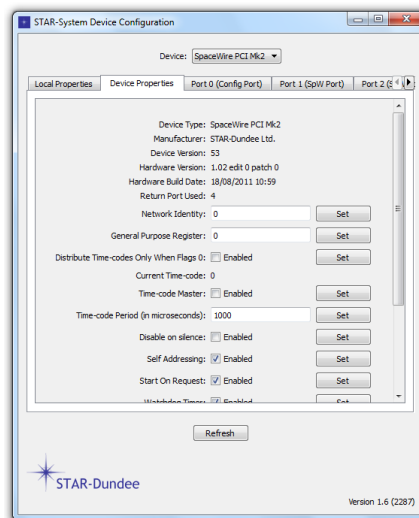
**Simple Tx GUI:** Allows a packet to be typed in and transmitted over the selected device and link. This makes it very easy to inject packets in to a network.

**Simple Rx GUI:** Allows packets to be received on the selected device and link, and the contents of the received packets to be displayed as a sequence of bytes. This is a very useful application for looking at the contents of small packets received on a network.

**Advanced Tx GUI:** Provides the ability to inject packets in to a network at very high data rates, specifying the format of packets, and the schedule to which the packets should be transmitted. The transmission statistics (shown below) can also be viewed, to monitor the rate at which packets are being transmitted.
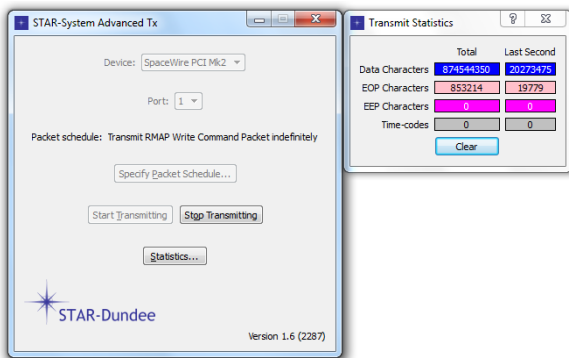


**Advanced Rx GUI:** Can be used to received packets at very high data rates, with the option to view the receive statistics and determine the rate at which packets are being received.

## Key Features

**Extensive API:** Developed after many years supporting users of SpaceWire interface boards, the STAR-System API provides a common API to STAR-Dundee's latest range of products, enhancing reusability of application software. The extensive functionality of the API makes it easy to develop SpaceWire related application software. Example application programs further relieve the pressure on development schedule. The API provides access to all the functionality of supported devices, providing the high performance data transfer capabilities often needed for SpaceWire system simulation and testing purposes.

**High performance:** STAR-System has been designed to provide the highest possible performance with the lowest possible CPU usage. STAR-Dundee has a wealth of experience developing high performance APIs and drivers, and this knowledge has been used to ensure STAR-System is capable of transmitting and receiving traffic at rates of multiple Gbits/s on a single link. This is achieved while keeping CPU usage to a minimum, so performance can be scaled to multiple links. Whilst SpaceWire links aren't capable of matching this performance, these rates ensure STAR-System can be used with SpaceFibre devices in the future, without requiring changes to user application software.

**Modularity:** In order to minimise the footprint of STAR-System, it is possible to customise the modules in the system that are included on the target hardware. For example, if a system does not require support for RMAP packets, the RMAP Packet Library does not need to be included on the target system.

**Ease of use:** STAR-System is designed to be easy to use. By taking advantage of features of each programming language, the API should be intuitive to use for anyone familiar with that language. Comprehensive documentation is also provided, including example source code demonstrating the use of each function.

**Comprehensive examples:** An example application is provided with source code, demonstrating the features provided by the API. This example code can be used as is, or can provide a reference point when writing your own application. An example application is also provided which can be used to test the performance of a device and/or the system. The source code for this application demonstrates potential ways in which to achieve the highest possible performance using the system.

**GUI applications:** Graphical applications are provided to configure devices, transmit packets and receive packets. These applications make it easy to start working with SpaceWire, while their flexibility means that they may be all you require to perform the tests, simulation or any other SpaceWire tasks.

**First class support:** As with all of our products, STAR-System includes a year's support and maintenance. This support is provided by the team that developed the software so we can respond quickly with answers to your questions, give assistance with application development, and quickly resolve any problems.

## Specifications

QNX Version supported: 6.5

Supported Languages and Environments:
- C.
- C++ (future addition, C API can be used in C++).

Supported Devices:
- SpaceWire PCI Mk2, cPCI Mk2.
- SpaceWire PCIe.

Provided APIs and Modules:
- STAR-API.
- RMAP Packet Library.
- Modules for Configuring Devices:
  - Remote Device Configuration API.
  - Router Configuration API.
  - Mk2 Interface and Router Configuration API.
  - SpaceWire PCI/cPCI Mk2 Configuration API.
  - SpW Router Mk2 Config. API (Q4 2012).
  - SpW-USB Brick Mk2 Config. API (Q4 2012).
- Drivers for individual device types:
  - SpaceWire PCI/cPCI Mk2 and PCIe Device Driver.

GUI Applications Included:
- Device Configuration GUI.
- Simple Tx GUI.
- Simple Rx GUI.
- Advanced Tx GUI.
- Advanced Rx GUI.

If you have a requirement for another language, operating system or device, please contact us and we will do our best to assist.